

## **Bilgisaymsal Düşünme Becerilerinin Oyun Programlama Aracılığıyla Geliştirilmesi: Ortaokul Öğrencileri için Bir Çerçeve**

Büşra ÖZMEN YAĞIZ  
Fırat Üniversitesi  
bozmen@firat.edu.tr  
ORCID ID: 0000-0003-3761-6215

Yasemin KOÇAK USLUEL  
Hacettepe Üniversitesi  
kocak@hacettepe.edu.tr  
ORCID ID: 0000-0002-6147-3333

Araştırma Makalesi	DOI: 10.31592/aeusbed.1444312
Geliş Tarihi: 28.02.2024	Revize Tarihi: 09.07.2024
	Kabul Tarihi: 30.07.2024

### **Atf Bilgisi**

Özmen Yağız, B. ve Koçak Usluel, Y. (2024). Bilgisaymsal düşünme becerilerinin oyun programlama aracılığıyla geliştirilmesi: Ortaokul öğrencileri için bir çerçeve. *Ahi Evran Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 10(2), 467-486.

### **ÖZ**

Bu çalışmanın amacı, bilgisayar oyunu programlama aracılığıyla bilgisayar düşünme becerilerinin geliştirilmesi ve değerlendirilmesi için öğretmenlere bir çerçeve sunmaktır. Bu çerçeve, ortaokul düzeyindeki öğrenciler için tasarlanmış bir dizi bilgisayar oyunu programlama etkinliğinden oluşmaktadır. Çerçevenin geliştirme süreci, durum çalışması yöntemiyle biçimlendirilmiştir. Bu doğrultuda alanyazın taraması, ihtiyaç analizi, öğrenen analizi ve doküman incelemesi yapılmıştır. Alanyazından hareketle, bilgisayar düşünme becerileri bağlamında; parçalara ayırma, örüntü tanıma, soyutlama, algoritma tasarımı ve hata ayıklama incelenmiştir. Bilgisaymsal düşünmede problem çözüme yaklaşımları olarak ise deneyimleme, üretme, hata ayıklama, azimli olma ve işbirliği yapma ele alınmıştır. İhtiyaç ve öğrenen analizleri kapsamında bilişim teknolojileri alanında en az beş yıllık tecrübeye sahip olan dört öğretmen ile yarı yapılandırılmış görüşmeler gerçekleştirilmiştir. Doküman incelemesi için ise farklı ülkelerin öğretim programlarının, ulusal ve uluslararası kuruluşların yönerge ve raporlarının bilgisayar düşünme becerileri açısından ayrıntılı bir değerlendirilmesi yapılmıştır. Araştırmacılar tarafından oluşturulan görüşme formları ve ulaşılan dokümanlardan toplanan verilerin çözümlenmesinde içerik analizinden faydalanılmıştır. Sonuç olarak, ortaokul öğrencilerine uygun öğrenme aktiviteleri ve kazanımlar içeren bir çerçeve geliştirilmiştir. Bu çerçevenin, öğrencilerin bilgisayar oyunu programlama ve bilgisayar düşünme becerilerinin geliştirilmesine yönelik düzenlenebilecek etkinlikler ve bu etkinliklerin öğretim ortamlarında nasıl uygulanacağı konusunda katkı getirebileceği ileri sürülebilir.

**Anahtar Kelimeler:** Bilgisaymsal düşünme, bilgisayar oyunu programlama, bilgisayar bilimi eğitimi, ortaokul öğrencileri, öğretim programı.

## **Developing Computational Thinking Skills through Computer Game Programming: A Framework for Middle School Students**

### **ABSTRACT**

This study aims to provide educators with a framework for the development and assessment of computational thinking skills through computer game programming. The framework consists of a series of computer game programming activities designed for students at the middle school level. The case study method was employed in the development of the framework. In this context, a literature review, needs analysis, learner analysis, and document analysis were conducted. Based on the literature, in the context of computational thinking skills; decomposition, pattern recognition, abstraction, algorithm design, and debugging were examined as computational thinking skills. Additionally, tinkering, creating, debugging, persevering, and collaborating were considered as problem-solving approaches in computational thinking. As a part of the needs and learner analyses, semi-structured interviews were organized with four teachers with at least five years of experience in the field of information technologies. For the document analysis, a detailed evaluation of the curricula of different countries, guidelines, and reports from national and international organizations was carried out in terms of computational thinking skills. Content analysis was used to analyze the data collected from the documents accessed and the interview forms developed by the researchers. As a result, a framework has been developed that includes learning activities and outcomes for middle school students. This framework has the potential to enhance the computational thinking and computer game programming skills of middle school students, as well as provide guidance on how these activities can be implemented in educational environments.

**Keywords:** Computational thinking, computer game programming, computer science education, middle school students, curriculum.

## Giriş

Bilgisaymsal düşünme (BD), bilgisayar bilimine ilişkin temel kavramları kullanarak insan davranışını anlama, sistem tasarımı ve problem çözmeyi kapsamaktadır (Wing, 2006). Bu bağlamda, bilgisayar bilimi, BD'nin geliştirilmesinde önemli bir disiplin olarak öne çıkmaktadır (Lodi ve Martini, 2021). Öğrencilerin gelecekteki yaşamlarında ihtiyaç duyacakları 21. yy. becerileriyle yetiştirilmesi için, bilgisayar biliminin (Herrero-Alvarez, Miranda, Leon ve Segredo, 2022) ve BD'nin (Grover ve Pea, 2018; Rich, Spaepen, Strickland ve Moran, 2019) temel eğitim programlarına entegrasyonu önemli bir öğrenme hedefidir. Bu kapsamda, farklı ülkeler BD'yi kritik bir beceri olarak kabul ederek öğrencilerin problem çözme, mantıksal düşünme ve algoritmik yaklaşımlar geliştirmelerine odaklanan programlar geliştirmektedir. Örneğin Birleşik Krallık (Department for Education [DE], 2013) ilkokuldan başlayarak programlamayı zorunlu bir ders olarak eğitim sistemine eklemiş, Finlandiya (Finnish National Board of Education [FNBE], 2016), Avustralya (Australian Curriculum, Assessment and Reporting Authority [ACARA], 2015), Danimarka (Tuhkala, Wagner, Iversen ve Kärkkäinen, 2019), Yeni Zelanda (New Zealand Ministry of Education [NZME], 2017), İsrail (Webb vd., 2017), Asya ülkeleri (örn. Güney Kore ve Çin) (Belmar, 2022) ve Türkiye (Milli Eğitim Bakanlığı [MEB], 2018) gibi ülkeler ise temel eğitim düzeyindeki öğretim programlarına bilgisayar bilimini ve BD'yi dahil etmişlerdir. Bununla birlikte, Bilgisayar Bilimi Öğretmenleri Derneği (Computer Science Teacher Association [CSTA], 2017) ve Uluslararası Eğitim Teknolojileri Derneği (International Society for Technology in Education [ISTE], 2017) BD'nin eğitim ortamlarında kullanımına ilişkin farklı yaklaşımları içeren standartlar ve kaynaklar geliştirmiştir.

Programlama eğitimi, BD'nin geliştirilmesinde tercih edilen yöntemler arasında dikkat çekmektedir (Tang, Yin, Lin, Hadad ve Zhai, 2020). Alanyazında programlamanın BD becerilerinin geliştirilmesinde en etkili yöntemlerden biri olduğu vurgulanmaktadır (Voogt, Fisser, Good, Mishra ve Yadav, 2015). Ancak, bazı kuramsal çalışma ve uygulamalarda BD'nin sadece programlama bağlamında ele alındığı görülmektedir (Denning, 2017). Bu durum; BD'ye, BD'yi değerlendirme yöntemlerine ve yürütülen programlama etkinliklerine ilişkin açıklayıcı bir betimlemenin yapılmasını güçleştirirken (Berland ve Lee, 2011) aynı zamanda BD'nin programlama öğretimini gerektirdiği gibi bir yanlı tavrın oluşmasına sebep olmaktadır. Bazı çalışmalarda, BD için programlamanın gerekli olduğu ifade edilse de (örn. Grover ve Pea, 2018; Laura-Ochoa ve Bedregal-Alpaca, 2022), temelde BD ile programlama eş kavramlar olarak kabul edilmemektedir. Wing (2006), BD'nin programlama dili okuryazarlığından fazlasına odaklandığını vurgulamaktadır. Bu kapsamda, BD yalnızca bilgisayar biliminin bir parçası olmakla kalmamakta, aynı zamanda farklı disiplinlerde bir araç olarak da kullanılabilir. Bununla birlikte, BD'nin temel amacı, problemlerin çözümünde ortak becerileri kullanabilme ve farklı disiplinlerde yeni problemler keşfedebilmedir (Barr ve Stephenson, 2011).

## Bilgisaymsal Düşünme

BD, bilgisayar bilimleri alanında temel bir kavram olarak kabul edilmektedir (Wing, 2006). Wing'e (2008) göre BD, karmaşık problemleri analiz etme ve çözme süreçlerinde algoritmaların ve mantıksal düşünmenin kullanımını içermektedir. Ayrıca okuma, yazma veya temel matematik gibi tüm bireylerin sahip olması gereken temel 21. yy. becerilerinden biridir ve bu nedenle temel eğitim programlarına dahil edilmelidir (Wing, 2006). Bununla birlikte, ISTE (2016) 21. yy. da temel eğitim öğrencilerinin geliştirmesi beklenen becerilere yönelik standartlar geliştirmiştir. "Bilgisaymsal düşünür", teknolojik bilgi ve becerilerinin geliştirilmesine yönelik yedi temel standarttan biridir. Bilgisaymsal düşünürler, karmaşık problemlerin tanımlanması ve çözülmesi sürecinde bilgi-işlem teknolojilerinden faydalanarak çeşitli stratejiler geliştiren ve bunları etkin bir biçimde kullanabilme becerisine sahip bireyler olarak ifade edilmiştir (ISTE, 2016). Wing (2008), BD'nin sadece problem çözenin değil, problemleri tanımlama ve geliştirmenin de merkezinde olduğunu, başka bir deyişle BD'nin makinelere ihtiyaç duymadığını, ancak insanların makineleri manipüle ederek BD süreçleri üretebileceğini vurgulamıştır. Bu nedenle, BD'nin sadece bilgisayar bilimi alanındaki öğrenciler için değil tüm öğrenciler için vazgeçilmez olduğunu belirtmiştir. Bu bağlamda BD, bilgisayar biliminin yanı sıra diğer disiplinlerde de çeşitli bağlamlarda problemlerin çözümünde prosedürel ve mantıksal düşünmeyi işe koşmak için kullanılabilen bir beceri seti olarak öne plana çıkmaktadır.

## Bilgisayımsal Düşünme Becerileri

BD becerileri, alanyazında farklı çalışmalarda değişkenlik göstermektedir. Ancak, bir alt küme içerisinde tekrarlanan bazı beceriler öne çıkmaktadır. Alanyazında geniş etkiye sahip, alıntılanma sayısı fazla olan ve CSTA ve ISTE gibi uluslararası alandaki organizasyonlara farklı bakış açıları sunan çalışmalardaki (Angeli vd., 2016; Barr ve Stephenson, 2011; Berland ve Lee, 2011; Grover ve Pea, 2018; Selby ve Woollard, 2013; Yadav, Mayfield, Zhou, Hambruch ve Korb, 2014) BD becerileri incelendiğinde, Wing'in çalışmalarında vurgulanan beceriler ile örtüştüğü görülmektedir. Buna göre;

- Soyutlama, BD'de en üst düzey ve önemli düşünme sürecidir (2011, s.1; 2017, s.8).
- BD ile karmaşık ve büyük bir görevin gerçekleştirilmesinde veya bir sistemin tasarlanmasında parçalara ayırma ve soyutlama kullanılır (2006, s.33).
- Algoritma, girdileri alarak bir dizi adımın sırasıyla uygulandığı ve çözüme ulaşmak üzere çıktıların üretildiği süreçlerin soyutlamasıdır (2011, s.1).
- Örüntülerin tanımlanmasında ve benzerliklerden yola çıkılarak genelleme yapılmasında soyutlama kullanılır (2011, s.1).
- Bilgi işleme, soyutlamanın otomasyonudur. Soyutlama (problemin çözümünde gerekli olan bilişsel araç), soyutlama katmanı (farklı düzeylerde çözülmesi gerekli olan problemler) ve bunlar arasındaki ilişkiler otomatikleştirilerek çalışılır. Otomasyon için açık sembol ve modellere ihtiyaç vardır (2008, s.3718).

Alanyazın ve Wing'in (2006; 2008; 2011; 2017) çalışmalarından derlenen kavramlar, BD'nin açık bir tanımlamasına işaret etmektedir. Bu noktada, BD'nin parçalara ayırma, soyutlama, algoritmik düşünme, otomasyon, örüntü tanıma ve genelleme ve değerlendirme becerilerini içerdiği söylenebilir. Bu çalışmada, alanyazında öne çıkan ortak becerileri dikkate alan Angeli vd. (2016) ve Csizmadia vd.'nin (2015) çalışmaları referans alınmıştır. Şekil 1, çalışmada ele alınan BD becerilerini ve tanımlarını içermektedir.

<b>Parçalara Ayırma</b>	Karmaşık bir problemin, anlaşılmasını ve çözümünü kolaylaştıracak küçük parçalara ayrılması (Csizmadia vd., 2015; Wing, 2006)
<b>Örüntü Tanıma</b>	Farklı problemlerde kullanılabilmesi için, çözümün içerdiği benzerlikleri bulma ve bunları genel terimlerle formülize etme (Csizmadia vd., 2015)
<b>Soyutlama</b>	Bir nesne veya varlıktan gereksiz niteliklerinin veya özelliklerinin, onu bir dizi ana özelliğe indirgemek için çıkarılması (Wing, 2011)
<b>Algoritma Tasarımı</b>	Problemin çözüm adımlarını içeren bir dizi talimat oluşturma (Csizmadia vd., 2015)
<b>Hata Ayıklama</b>	Çözümdeki hataları bulma ve düzeltme (Berland ve Lee, 2011; Csizmadia vd., 2015)

Şekil 1. BD Becerileri ve Tanımları

Anderson (2016), BD'nin, her biri problemin çözümünde yararlı olan bu beş temel becerinin aşamalı olarak bir araya gelmesini içerdiğini belirtmiştir. Bu anlayışla, ilk olarak çözülecek karmaşık problem yönetilebilir küçük parçalara ayrılmaktadır. Ardından, tanımlanan problem parçaları içinde bir çözümün daha etkili bir biçimde tasarlanmasında yararlanılabilecek örüntüler belirlenerek modeller oluşturulmaktadır. Daha sonra, bu modeller gereksiz ayrıntılarından arındırılarak tümel bir biçimde sunulmakta, başka bir deyişle soyutlanmaktadır. Soyutlama ile ulaşılan çözümün, sistematik olarak ortaya konulması için bir dizi adımdan oluşan algoritması geliştirilmektedir. Son olarak, bütüncül bir çözümün gerektirdiği aşamaların tümünün değerlendirilmesi ve verimliliğin artırılması amacıyla geliştirilen algoritma, süreç veya sistem değerlendirilmektedir.

Bir problemin karmaşıklığını azaltmak için genellikle problemi daha anlaşılır ve çözülebilir küçük parçalara ayırmak gerekmektedir (Anderson, 2016; Curzon, Dorling, Ng, Selby ve Woollard, 2014). Parçalara ayırma, alt problemler ile bunları çözebilmek için gereken yöntemlerin tanımlanmasını

çermektedir (Barr ve Stephenson, 2011). Bu yaklaşım, yeni durumların anlaşılmasını, büyük ve karmaşık sistemlerin tasarımını ve problemlerin çözümünü kolaylaştırmaktadır (Csizmadia vd., 2015). Ayrıca, çözüm sürecini daha yönetilebilir kılmaktadır. Parçalara ayırma aynı zamanda, problemin, her biri farklı iç görüler, beceriler ve deneyimleri bir araya getiren takım arkadaşları tarafından ele alınmasını desteklemektedir (Computing at School [CAS], 2014).

Örüntü tanıma, daha önce formüle edilmiş problemleri ve bunların çözümlerini yeni problemlere uyarlama sürecidir (Curzon vd., 2014). Problemler ve çözümler içerisinde tekrarlayan her türlü bilgiyi temsil eden kalıplar ve modellerin oluşturulmasını içermektedir (Anderson, 2016). Örüntü tanıma sırasında, "Daha önce çözdüğüm bir problem ile benzerlik gösteriyor mu?", "Çözdüğüm problemler ile bağlantılar içeriyor mu?", "Önceden kullandığım problem çözme stratejileri, bu probleme transfer edilebilir mi?" benzeri sorulara yanıt aranmaktadır (Csizmadia vd., 2015, s.8). Bu beceri, deneyimlerden faydalanılarak problem çözme sürecinin optimize edilmesine ve böylece yeni problemlerin daha hızlı, etkili ve işlevsel bir biçimde çözülmesine olanak tanımaktadır (Csizmadia vd., 2015). Aynı zamanda, örüntü tanıma ile örüntüler belirlenerek tahminler yapılabilmekte, kurallar oluşturabilmekte ve daha geniş kapsamlı problemlere çözüm üretilebilmektedir (CAS, 2014).

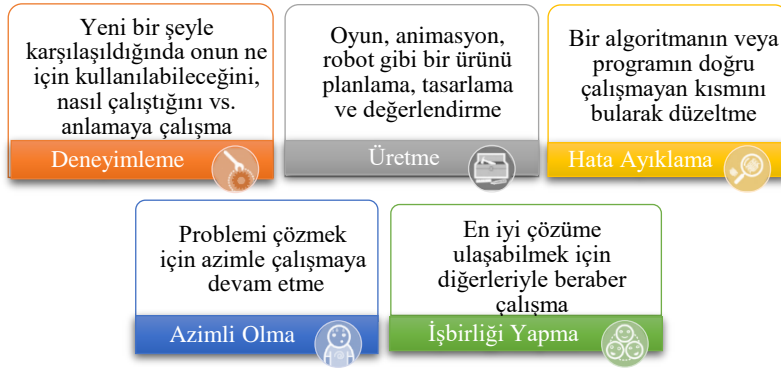
Soyutlama, problemleri çözmek için geliştirilen bir modelin ayrıntılarının filtrelenerek genelleştirilmiş bir temsil oluşturulması sürecidir (Anderson, 2016). Bu aşamada, modelin doğru bir temsilinin seçilmesi oldukça önemlidir (Csizmadia vd., 2015). Soyutlama, bilgiyi kategorize etme, modelleme ve genellemeyi sağlamaktadır. Böylece yeni bilgiyi daha hızlı entegre etmeyi, daha etkili kararlar almayı ve karmaşıklığın yönetilmesini kolaylaştırmaktadır (Csizmadia vd., 2015; Waite, Curzon, Marsh, Sentance ve Hadwen-Bennett, 2018). Bir başka deyişle, soyutlama ile belirli detayların göz ardı edilerek temel ilişkilerin ve yapıların vurgulanması amaçlanmaktadır (örn. program yazarken belirli bir işlevin nasıl gerçekleştirileceğine odaklanılırken, soyutlama yapılarak bu işlevin karmaşıklığı azaltılabilir ve genel probleme odaklanılabilir). Wing (2011; 2017), BD'nin merkezinde yer alan becerinin soyutlama olduğunu ve hangi ayrıntıların vurgulanacağı, hangilerinin dikkate alınmayacağına karar vermeyi içeren bir süreç olduğunu belirtmiştir.

Algoritma, bir problemin çözümü için üretilen komut veya kurallar dizisi olarak tanımlanmaktadır (Futschek, 2006). Algoritmik düşünme ise problemi analiz etme, uygun çözüm stratejilerini belirleme, bu stratejilere dayalı adımları sıralama ve son olarak algoritmayı doğru bir şekilde uygulama süreci olarak ifade edilmektedir (Csizmadia vd., 2015). Papert (1980), bilgisayar bilminde algoritmik düşünmeyi bireylerin kendi programlarını geliştirirken öğrendikleri temel bir beceri olarak vurgulamaktadır. Algoritma tasarımı, problem çözümünde doğru ve etkili bir yöntem geliştirmek için soyut düşünme yeteneğini gerektirmektedir (Wing, 2011). Bu nedenle, BD'nin temel bir unsuru (Israel, Pearson, Tapia, Wherfel ve Reese, 2015) ve çeşitli alanlardaki problemlerin çözümünde yararlanılabilecek güçlü bir araç olarak değerlendirilmektedir.

Probleme iyi bir çözümün geliştirilmesinde, çözümün amaca uygunluğunun değerlendirilmesi gerekmektedir (Liu, Zhi, Hicks ve Barnes, 2017). Hata ayıklama, değerlendirmenin bilgisayar bilmindeki karşılığı olarak ele alınmakta ve sistemlerin veya programların beklenmeyen davranışlarını tanımlama ve düzeltme süreci olarak betimlenmektedir. Bu süreç, yazılım geliştirmenin ayrılmaz bir parçası ve bir sistem veya programın doğru çalışması için gerekli bir adımdır. Hata ayıklama ile çözüm, gereksiz adımlar (verimlilik), gerekli tüm adımlar (kapsamlılık), hatalı adımlar (kullanılabilirlik ve etkililik) gibi özellikleri açısından değerlendirilmektedir (Anderson, 2016). Aynı zamanda, mevcut kurallar ve/veya stratejiler incelenmektedir. Papert'e (1980) göre hata ayıklama, prosedürel düşünmenin en "güçlü fikri"dir. Berland ve Lee (2011) ve Wing (2006) ise hata ayıklamayı, eleştirel ve algoritmik düşünmeyi kapsadığı için hem BD hem de programlama için kritik bir beceri olarak tanımlamışlardır.

## **Bilgisaymsal Düşünme Yaklaşımları**

Woollard (2016), deneyimleme (keşfetme), tasarlama ve üretme, hata ayıklama, azimli olma ve işbirliği yapma olmak üzere beş problem çözme yaklaşımının BD becerilerinin gelişiminde sıklıkla kullanıldığını ifade etmiştir (s.5). Şekil 2'de söz konusu yaklaşımlar ve tanımları sunulmuştur.



Şekil 2. BD'de Problem Çözme Yaklaşımları

Deneyimleme, bir kişinin yeni bir şeyle karşılaştığında, onun işlevini, özelliklerini, çalışma biçimini vs. anlamaya çalışması bir başka deyişle keşfetmesi olarak tanımlanmıştır (örn. yeni alınan bir telefonun tüm fonksiyonlarını incelemek ve kullanışlı olanlarını belirlemek) (Blikstein, Worsley, Piech, Sahami, Cooper ve Koller, 2014). Deneyimleme, yeni şeyler denedikçe neden-sonuç ilişkisi kurma imkânı sunmaktadır. Bu nedenle, mantıksal akıl yürütmeye ilişkilendirilmektedir (Kim, Belland, Baabdullah, Lee, Dinç ve Zhang, 2021). Üretme, bir projedeki oyun, animasyon gibi bir ürünü planlama, tasarlama ve değerlendirme becerilerini kapsamaktadır (örn. oyun programlama aracılığıyla içerik tasarlamak ve geliştirmek) (Mannilla Leinonen, Bauters ve Veermans, 2023). Papert'e (1980) göre, yaratıcı bir süreç olduğu dikkate alındığında, bilgi işleme sürecinde özgün, değerli ve aynı zamanda geniş çapta etki yaratan ürünlerin geliştirilmesi büyük önem taşımaktadır. Hata ayıklama, bir algoritmanın veya programın doğru çalışmayan kısmını bularak düzeltme olarak tanımlanmaktadır (Csizmadia vd., 2015). Programlamanın önemli bir parçası olan hata ayıklama, bireylerin günlük problemleri çözerken benzer bir yaklaşımı benimsemeleri nedeniyle günlük yaşamda da sıklıkla kullanılmaktadır. Bu bağlamda, karşılaşılan bir problemi tanımlama, analiz etme, nedenlerini belirleme ve uygun bir çözümü uygulamayı içermektedir (örn. bir cümlenin dilbilgisi açısından doğruluğunu kontrol etmek ve düzeltmek). Azimli olma, problemi çözmek için vazgeçmeden, azimle çalışmayı sürdürme olarak tanımlanmıştır (örn. algoritma veya program geliştirirken uzun süre çalışmak ve sürekli tekrarlar yapmak) (Wahyuni, Rozimela, Ardi, Mukhaiyar ve Darmansyah, 2022). Bu yaklaşım, aynı şeyi defalarca tekrarlamak yerine yeni şeyler öğrenirken kendi öğrenme potansiyelini zorlamak olarak da ifade edilmektedir (Woollard, 2016). Başarılı programlar geliştirmek için yalnızca programlama kavramlarını bilmek veya algoritma mantığını kavramak yeterli görülmemektedir (Hunsaker, 2018). Bunun için, algoritma oluşturmaya ve program yazmaya oldukça fazla çaba ve zaman harcamak ve zor problemleri bile çözmeye istekli olmak gerekmektedir. İş birliği yapma, en iyi çözüme ulaşabilmek için diğerleriyle beraber çalışmadır (Woollard, 2016). Tek başına çalışıldığında başa çıkılamayacak seviyede karmaşık olan görevlere devam etmek için gerekli motivasyonu sağlamakta ve başarıyı paylaşma fırsatı sunmaktadır (örn. bir video oluştururken grup üyelerinin projeyi içerik geliştirme, tasarım vs. gibi bölümlerine ayırarak her bir üyenin farklı roller üstlenmesi) (Manilla vd., 2023).

Alanyazında, programlama okuryazarlığının ötesinde BD becerilerinin ve yaklaşımlarının değerlendirildiği çalışmaların yapılmasına ihtiyaç olduğu dile getirilmektedir (Lockwood ve Mooney, 2018; Pelanek ve Effenberger, 2023). Bununla birlikte, öğretim etkinliklerinin geliştirilmesinde, mevcut öğretim programlarıyla ilişkilendirilmiş ve bu etkinliklerin öğretim ortamlarında nasıl uygulanacağına dair uygulayıcılara rehberlik edecek yeterli öneri bulunmadığı da göze çarpmaktadır. Bu araştırmada, ortaokul öğrencilerinin programlama becerileri ile birlikte BD becerilerinin, bilgisayar oyunu programlama etkinlikleri aracılığıyla geliştirilmesi ve değerlendirilmesinde örnek bir çerçevenin oluşturulması amaçlanmıştır. Temel eğitim düzeyinde BD becerilerinin geliştirilmesi amacıyla geliştirilen bu çerçevenin ulusal öğretim programlarına da katkı sağlayacağı öngörülmektedir.

## Yöntem

Bu çalışmada, ortaokul öğrencilerinin BD becerilerini geliştirmeye yönelik ihtiyacın belirlenmesi, hedef kitlenin hazırbulunuşluluğunun değerlendirilmesi ve mevcut öğretim

programlarındaki durumun betimlenmesi amacıyla durum çalışmasından faydalanılmıştır. Durum çalışması, "nasıl" ve "neden" sorularını ele alarak belirli bir olayı, durumu veya olguyu çok yönlü ve derinlemesine incelemek için kullanılan bir araştırma yöntemidir (Yin, 2014). Bu kapsamda, çalışmada hangi BD becerilerinin temel alınacağına karar verilmesinde derinlemesine bir alanyazın taraması yapılmıştır. Tasarlanması planlanan öğretim etkinliklerine ilişkin ihtiyacın ve olası çıktıların tanımlanması için ihtiyaç analizi gerçekleştirilmiştir. Ayrıca, hedef kitlenin mevcut durumunun ve öğretim etkinliklerinin hedef kitleye uygunluğunun belirlenmesi amacıyla öğrenen analizi yürütülmüştür. Mevcut öğretim programlarında BD becerilerine ne düzeyde ve şekilde yer verildiğinin ve BD'ye ilişkin hangi kazanımların yer aldığının tespit edilmesi amacıyla ise çeşitli makaleler, raporlar ve Türkiye'nin ve diğer ülkelerin öğretim programları doküman analizi aracılığıyla incelenmiştir. Analizler sonucunda belirlenen ana başlıklardan yola çıkılarak ortaokul öğrencilerinin BD becerilerinin geliştirilmesi ve değerlendirilmesinde kullanılabilecek bir öğretim programı önerisi geliştirilmiştir.

## **Katılımcılar**

Araştırmanın katılımcılarını, Türkiye'nin doğusundaki bir ilin merkezinde görev yapmakta olan ve bilişim teknolojileri alanında uzman olan dört öğretmen oluşturmuştur. Katılımcıların seçilmesinde ölçüt örnekleme kullanılmıştır. Bu yöntem, katılımcıların belirlenen ölçütlere göre seçilmesine dayanmaktadır (Patton, 2014). Bu doğrultuda, ortaokul düzeyinde "Bilişim Teknolojileri ve Yazılım (BTY)" dersini veren ve alanında en az beş yıllık deneyimi olan öğretmenler seçilmiştir. Çalışmanın ortaokul düzeyinde yürütülmesinin temel nedeni, Türkiye'deki zorunlu BTY dersinin altıncı sınıf öğretim programında yer alan problem çözme, programlama ve ürün geliştirme becerileri ile özellikle ikinci dönem kazanımlarında yer alan oyun programlama becerilerinin bu çalışmanın odak noktası olan oyun programlama ve BD becerileri ile önemli derecede örtüşmesidir. Ayrıca, beşinci sınıf öğretim programının temel bilgisayar kullanımı ve programlama becerilerini içermesi, dolayısıyla öğrencilerin oyun programlama etkinlikleri için gerekli ön bilgiye sahip olmaları da bir diğer önemli faktördür.

## **Veri Toplama Araçları**

Veri toplama araçlarını görüşme formları ve dokümanlar oluşturmaktadır. Araştırmacılar tarafından, ihtiyaç analizi ve öğrenen analizlerinde kullanılması için yarı yapılandırılmış görüşme formları geliştirilmiştir. Doküman analizi için ise ortaokul düzeyinde BD becerilerinin geliştirilmesini içeren öğretim programları, yönergeler, raporlar ve makaleler incelenmiştir.

İhtiyaç analizi kapsamında gerçekleştirilen görüşmelerde bilişim teknolojileri öğretmenlerine dört soru yöneltilmiştir. Bunlar; BD becerilerinin geliştirilmesi için bilgisayar oyunu programlama etkinliklerine ihtiyaç olup olmadığı, çalışmada geliştirilecek etkinliklerin problem çözme, programlama, BD becerileri ve BD yaklaşımları açısından sağlayabileceği katkılar, öğretim sürecinden beledikleri çıktılar ve mevcut öğretim programını uygularken yaşadıkları sıkıntılar ile bunların çözümünde ihtiyaç duydukları pedagojik ve teknolojik destekleri içermektedir. Öğrenen analizi kapsamında dört soru sorulmuştur. Bunlar; öğretmenlerin BD'yi geliştirmek için daha önce hangi araçları kullandığı ve öğrencilerin öğretim programındaki programlama becerileri, BD becerileri ve BD yaklaşımlarına sahip olma düzeylerini içermektedir. Görüşme formları, uzman değerlendirmesi kapsamında, nitel araştırma yöntemleri konusunda uzman ve öğretim teknolojileri alanında çalışan bir akademisyen tarafından incelenmiş ve görüşleri doğrultusunda yeniden düzenlenmiştir.

Doküman analizi, basılı ve/veya elektronik dokümanların incelenmesini ve sistematik değerlendirilmesini içermektedir. Doküman analizi ile anlamın ortaya çıkarılması ve nihai olarak bilginin üretilmesi hedeflenmektedir (Bowen, 2009). Bu çalışmada, doküman analizi kapsamında öncelikle bilgisayar bilimi eğitimi ile birlikte, BD'yi öğretim programlarına dahil eden öncü ülkeler belirlenmiştir. Bu ülkeler; Birleşik Krallık (DE, 2013), Finlandiya (FNBE, 2016), Avustralya (ACARA, 2015), Yeni Zelanda (NZME, 2017), İsrail (Webb vd., 2017) ve Türkiye (MEB, 2018) olarak sıralanmaktadır. Daha sonra bu ülkelerin öğretim programları, BD'ye yer verme düzeyi ve biçimi açısından incelenmiştir. Ayrıca, ulusal ve uluslararası kuruluşların ilgili yönergelerine, BD'ye ilişkin uluslararası kuruluşların raporlarına ve bilimsel çalışmalara da ulaşılarak değerlendirilmiştir.

## Verilerin Analizi

Veriler içerik analizi ile çözümlenmiştir. Bu analiz yöntemi, nitel veriler kullanılarak verinin içeriğine ilişkin geçerli ve tekrarlanabilir çıkarımların yapılmasını içermektedir (Krippendorff, 2004). Görüşme verilerinin analizinde, Miles ve Huberman'ın (1994) önerdiği verilerin toplanması, özetlenmesi, görselleştirme, sonuç çıkarma ve teyit etme adımlarını içeren tümevarımsal veri analiz yaklaşımından faydalanılmıştır. Yüz yüze gerçekleştirilen görüşmeler sonucunda, öğretmenlerin sorulara verdikleri cevaplar araştırmacı tarafından not edildikten sonra elektronik ortama aktarılmıştır. Daha sonra, verileri iyice tanıyarak güçlü bir içgörü elde etmek için dikkatlice okunarak araştırma problemine dayalı notlar alınmıştır. Kodlama işlemi için, Strauss ve Corbin'in (1998) önerdiği açık kodlama ile serbest kod listeleri oluşturulmuştur. Benzerlik gösteren veriler sistematik bir biçimde gruplandırılarak tema ve alt temalar oluşturulmuştur. Görüşmelerden elde edilen bulgular alıntılarla birlikte tablolaraştırılarak sunulmuştur. Doküman analizi sürecinde ise Merriam'ın (2009) önerdiği, uygun dokümanları bulma, dokümanların özgünlüğünü kontrol etme, kodlama ve kataloglama için bir sistematik oluşturma ve veri analizini gerçekleştirme aşamaları izlenmiştir.

## Geçerlik ve Güvenirlik

Nitel araştırmalarda geçerlik ve güvenilirlik; inandırıcılık, aktarılabilirlik, tutarlılık ve teyit edilebilirlik kavramlarıyla ilişkilendirilmektedir (Lincoln ve Guba, 1985). Bu araştırmada, inandırıcılık ve teyit edilebilirliğin sağlanması için birden fazla veri toplama aracı kullanılarak veri üçgenlemesi yapılmıştır. Ayrıca veri toplama araçlarının hazırlama sürecinde uzman görüşlerinden faydalanılmıştır. Aktarılabilirliği sağlamak için azami çeşitlilik ve ayrıntılı betimleme stratejileri kullanılmıştır. Çalışma grubu, ortam ve süreçler ayrıntılı olarak anlatılmış ve çalışma grubu olarak özel ve devlet okullarından öğretmenler seçilmiştir. Tutarlılığı sağlamak amacıyla yine veri üçgenlemesi ve denetleme tekniği kullanılmıştır. Bu noktada, elde edilen bulguların tutarlılığı olduğu ve birbirini desteklediği görülmüştür. Bununla birlikte, okuyucunun araştırmayı adım adım takip edebilmesi için veri toplama süreçlerinde izlenen yollar gerekçeleriyle birlikte detaylı bir biçimde raporlanmıştır.

## Araştırma Etiği

Araştırma için etik onay, Hacettepe Üniversitesi Sosyal ve Beşeri Bilimler Araştırma Etik Kurulu'ndan (Onay numarası: 35853172/433-3453) alınmıştır. Ayrıca araştırmaya, verilerin toplandığı ilin Milli Eğitim Müdürlüğü (Sayı: 79137285-605-E.12854266) tarafından uygulama izni verilmiştir.

## Bulgular

### İhtiyacın ve Olası Çıktıların Belirlenmesine Yönelik Bulgular

Bu çalışmada, oyun programlama etkinlikleri aracılığıyla BD becerilerinin geliştirilmesine yönelik ihtiyacın ve olası çıktıların betimlenmesi için ihtiyaç analizi gerçekleştirilmiştir. Tablo 1'de analiz sonuçlarına dayanarak oluşturulan tema, alt tema ve doğrudan alıntılara örnekler verilmiştir.

Tablo 1

*İhtiyacın ve Olası Çıktıların Belirlenmesine İlişkin Temalar, Alt Temalar ve Doğrudan Alıntılar*

Temalar	Alt Temalar	Doğrudan Alıntılar	
Programlama Becerisi	Karar yapıları	"Öğrenciler, şart ifadelerindeki eğer yapısını daha iyi kavrar." (ÖĞRT1)	
	Doğrusal mantık	"Kodlamayla bir görevin adımlarını görmüş olurlar." (ÖĞRT4)	
BD Becerileri	Algoritma tasarımı	"İyi bir programlama için algoritmanın doğru oluşturulması çok önemli. Algoritma oluşturmayı öğrenebilirler." (ÖĞRT2)	
	Hata ayıklama	"Kendi yanlış kodlarını düzeltmeyi öğrenmeleri için faydalı olabilir." (ÖĞRT4)	
	Deneyimleme	"Farklı bir aracı tanımış olurlar, alternatif bir araç işe yarayacaktır." (ÖĞRT3)	
BD Yaklaşımları	Üretme	"Kendi üç boyutlu karakterlerini oluşturması, daha çok kendine ait oyun hazırlamasını sağlayacaktır." (ÖĞRT3)	
	İşbirliği	Yardımlaşma	"Problemin çözümü için arkadaşlarından yardım alabilir." (ÖĞRT1)
	Paylaşım	"Öğrenci işbirliğiyle veri toplayıp paylaşabileceğinin farkına varabilir." (ÖĞRT2)	
		"Oyununu arkadaşlarıyla paylaşabilme fırsatı da elde eder." (ÖĞRT3)	

Temalar	Alt Temalar	Doğrudan Alıntılar
Problem Çözme Becerisi	Farklı çözümler geliştirme	"Problemlere farklı farklı çözümler üretebilmelerini sağlayabilir diye düşünüyorum." (ÖĞRT2)
	Farkındalık oluşturma	"Günlük hayatta karşılaşılabilecekleri problemler hakkında farkındalıkları artacaktır." (ÖĞRT3)
	Çözüm üretme	"Somut öğrenme ortamında kendi projeleri üzerinde uzunca bir süre çalışırlarsa problemlere çözüm üretebilirler." (ÖĞRT2)
Duyuşsal Çıktılar	Motivasyon	"Kendi hikayelerini yazarak oyun yapması öğrenciyi daha çok motive edecektir." (ÖĞRT3)
	Özgüven	"Derste üretim yaptıklarında daha çok cesaretlenip kendi oyunlarını programlama konusunda çok daha özgüvenli hissetmelerine yardımcı olunabilir." (ÖĞRT2)
	İlgi	"Kendi oyunlarını programlamayı eğlenceli bulurlarsa derse daha çok katılırlar." (ÖĞRT1)
Bilişsel Çıktılar	Algı	"Bazı öğrenciler programlama etkinliklerine katılmak istemiyor, zor buluyor. Oyun programlama bu bakış açılarını değiştirebilir." (ÖĞRT2)
	Merak	"Programlama yapılarını öğrenmeleri için merak duygularının oluşturulmasına yardımcı olur." (ÖĞRT4)
	Hayal gücü	"Hikaye yazıp oyuna dökmeleri hayal güçlerini geliştirebilir." (ÖĞRT4)
Öğretim Programı	Süre	"Bir ders saatinde süre yetiremiyorum, farklı konulara geçmemiz gerekiyor, sonra unutupuyorlar, bir dönemlik proje olması işe yarayabilir." (ÖĞRT3)
	Değerlendirme aracı	"Sınavla sadece yazdıkları kodları değerlendirebiliyorum. Farklı değerlendirme araçları elimizde olsa daha kolay değerlendirebiliriz." (ÖĞRT1)
	Pedagojik yaklaşım	"Proje tabanlı öğretim gerçekleştirmemiz gerekiyor ama bunun için hem süre yok hem de öğrencilerin hepsi aynı bilgiye ve öğrenme hızına sahip değil." (ÖĞRT4)

Analiz sonuçlarına göre programlama becerisi, BD becerileri, BD yaklaşımları, problem çözme becerisi, duyuşsal çıktılar, bilişsel çıktılar ve öğretim programı şeklinde yedi tema oluşturulmuştur. Bunlar arasında en yoğun yanıtlan; BD yaklaşımları, BD becerileri ve öğretim programı konularında alınmıştır. Öğretmenlerin çoğunlukla, planlanan öğretim etkinliklerinin öğrencilerin deneyimleme, üretme ve işbirliğine yönelik ihtiyaçlarını karşılayacağı yönünde görüş bildirdikleri belirlenmiştir. Ayrıca işbirliği açısından yardımlaşma ve paylaşım önemli temalar olarak ortaya çıkmıştır.

BD becerileri bağlamında, hata ayıklama ve algoritma tasarımı becerileri öne çıkarken; programlama becerileri açısından karar yapıları ve doğrusal mantık temaları hakkında bildirilen görüşler göze çarpmaktadır. Bu sonuçlar, öğretmenlerin BD'yi programlama becerisiyle eş değer olarak algıladıkları şeklinde yorumlanabilir. Bu da, öğretmenlerin diğer BD becerileri hakkında bilgilerinin sınırlı olabileceğine işaret etmektedir. Problem çözme becerilerinde ise farklı çözüm yaklaşımları geliştirme, farkındalık oluşturma ve çözüm üretme öne çıkan alt temalardır. Öğretmenler, çoğunlukla BD öğretim etkinliklerinin problem çözme becerilerine olumlu katkı sağlayabileceğini düşünmektedirler.

Olası çıktılar bağlamında, duyuşsal ve bilişsel çıktılara dair yanıtlar verildiği kaydedilmiştir. Bilişsel açıdan algı, merak ve hayal gücüne vurgu yapılırken; duyuşsal açıdan motivasyon, özgüven ve ilgi temaları öne çıkmıştır. Bu sonuçlar, öğretmenlerin tasarlanacak öğretim programının öğrencilerde duyuşsal ve bilişsel açıdan olumlu etkiler yaratacağını düşündükleri şeklinde yorumlanabilir.

Öğretmenler mevcut öğretim programında süre, değerlendirme aracı ve pedagojik yaklaşımla ilgili eksiklikler ve zorluklar olduğunu ifade etmişlerdir. Buna göre, özellikle özgün bir proje üretimi aracılığıyla BD becerilerinin geliştirilmesi için programda ayrılan sürenin artırılması, uygulayıcılara kılavuzluk edebilecek değerlendirme yöntemlerinin geliştirilmesi ve öğrencilerin ihtiyaçlarına uygun öğrenme ortamlarının ve içeriğin hazırlanmasına ihtiyaç olduğu açıkça ortaya çıkmıştır. Özetle, bulguların programlama eğitimi aracılığıyla BD gelişiminde odaklanılması gereken ana hatları ortaya koyduğu ileri sürülebilir.

## Hedef Kitlenin Mevcut Durumunun Belirlenmesine Yönelik Bulgular

Tasarlanacak öğretim etkinliklerinin hedef kitleye uygunluğunun ve hedef kitlenin mevcut durumunun belirlenmesi için öğrenen analizi gerçekleştirilmiştir. Tablo 2'de, sonuçlar ışığında ortaya konulan temalar, alt temalar ve doğrudan alıntı örnekleri sunulmuştur.



Tablo 2

*Hedef Kitlenin Mevcut Durumunun Belirlenmesine Yönelik Temalar, Alt Temalar ve Doğrudan Alıntılar*

Temalar	Alt Temalar	Doğrudan Alıntılar
Programlama Becerisi	Değişkenler	"Değişken kavramını anlayamayan öğrencilerin sayısı çok." (ÖĞRT2)
	Karar yapıları	"Çoklu şart ifadelerinde önce hangi ifade ile başlayacaklarını karıştırıyorlar." (ÖĞRT2)
	Döngü yapıları	"Altıncı sınıflar en çok döngülerde zorlanıyor." (ÖĞRT4)
BD Becerileri	Hata ayıklama	"Kendi programlarındaki hataları bulmakta zorlandılar." (ÖĞRT3)
	Algoritma tasarımı	"Geçen dönemden Scratch biliyorlar ama algoritma oluştururken hala sorun yaşıyorlar." (ÖĞRT1)
BD Yaklaşımları	Üretme	"Derste uygulama yapmak için yeterli süre yok, o nedenle dönem sonunda proje geliştiremiyoruz." (ÖĞRT4)
	Deneyimleme	"Scratch ve Code.org'u kullandılar daha önce." (ÖĞRT1) "İlk dönem programlama için Scratch'i kullanıyoruz." (ÖĞRT4)

Analiz sonucunda, öğretmenlerin yanıtları programlama becerisi, BD becerileri ve BD yaklaşımları olmak üzere üç temada toplanmıştır. Programlama becerisi bağlamında öğrencilerin yoğunlukla değişkenler, karar yapıları ve döngü yapıları konularında eksikliklerinin olduğu belirlenmiştir. Bununla birlikte, BD becerileri açısından hata ayıklama ve algoritma tasarımı becerilerine ilişkin eksikliklerin ifade edildiği görülmüştür. Son olarak, tüm öğrencilerin daha önce bir görsel programlama aracını deneyimlediği tespit edilmiştir. Ayrıca, üretim odaklı bir çalışma yürütebilmek için yeterli süre ve kaynağın olmadığına dair çok sayıda görüş bildirildiği dikkat çekmiştir. Bu noktada, öğrencilerin görsel programlamaya aşına oldukları ancak özgün bir proje üretecek kadar programlama ve BD becerilerine sahip olmadıkları söylenebilir.

**Öğretim Hedeflerinin Belirlenmesine Yönelik Bulgular**

Öğretim hedeflerinin oluşturulması için, öncelikle çalışmada ele alınacak BD becerileri belirlenmiştir. Bu noktada, kuramsal temel ve ilgili araştırmalar yön gösterici olmuştur. Alanda öncü kabul edilen Wing'in (2006; 2008; 2011; 2017) araştırmaları, ISTE (2017) ve CSTA (2017) gibi uluslararası çalışma gruplarına çeşitli bakış açıları sunan, alanda geniş etkisi olan ve fazlaca alıntılanan çalışmalarda (Angeli vd., 2016; Berland ve Lee, 2011; Csizmadia vd., 2015; Grover ve Pea, 2018; Selby ve Woollard, 2013; Yadav vd., 2014) becerileri değerlendirilmiş, sonuçta Angeli vd. (2016) ve Csizmadia vd. (2015) tarafından önerilen beş beceri temel alınmıştır; parçalara ayırma (PA), örüntü tanıma ve genelleme (ÖG), soyutlama (SY), algoritma tasarımı (AT) ve hata ayıklama (HA).

BD becerilerine karar verilmesinden sonra, Türkiye'de mevcut öğretim programlarında BD'ye yer verilme düzeyinin ve şeklinin belirlenmesi açısından BTY dersinin öğretim programı derinlemesine incelenmiştir. Sonuçta; programlama, üretme ve problem çözme becerilerinin öğretim programında belirgin bir yer tuttuğu ve altıncı sınıf düzeyinde ikinci dönemin tamamında bu becerilerin geliştirilmesine odaklanıldığı görülmüştür. Ek olarak, BTY dersinin öğretim programının programlama, algoritma tasarımı, problem çözme, BD becerileri, gerçek yaşam problemlerinin çözümü için özgün proje tasarımı ve üretmeyi kapsayan genel ve özel amaçlarının, araştırmanın amaçları ile büyük oranda örtüştüğü de dikkat çekmiştir (MEB, 2018, s.5).

Türkiye'den sonra, farklı öğretim düzeylerinde bilgisayar bilimini ve BD'yi öğretim programlarına dahil eden diğer öncü ülkelerin öğretim programları incelenmiştir. Bu ülkeler; Birleşik Krallık (DE, 2013), Finlandiya (FNBE, 2016), Avustralya (ACARA, 2015), Yeni Zelanda (NZME, 2017) ve İsrail (Webb vd., 2017) olarak sıralanmaktadır. İnceleme sonucunda, tüm ülkelerin ortak olarak algoritmalar veya algoritmik düşünme ve programlama veya kodlama becerilerine; Birleşik Krallık, İsrail, Finlandiya ve Avustralya'nın problem çözme becerisine; Birleşik Krallık'ın veri gösterimi, parçalara ayırma, hata ayıklama ve soyutlama becerilerine ve son olarak Yeni Zelanda, İsrail, Finlandiya ve Avustralya'nın veri yönetimi ve analizi becerilerine programlarında yer verdiği görülmüştür.

Yeni kazanımların yazılması için, öğretim programlarından sonra farklı kaynaklar da BD becerileri açısından analiz edilmiştir. İlk olarak, CSTA (2017) ve ISTE (2017) tarafından hazırlanan raporlar incelenmiştir. Bu raporlarda, farklı öğrenim düzeylerinde BD becerilerinin desteklenmesine

ilişkin kazanımlar sunulmuştur. Raporlarda odaklanılan BD becerileri; veri toplama, çözümlleme, ve sunma, problemi parçalarına ayırma, algoritma ve prosedürler, otomasyon, soyutlama, paralelleştirme ve simülasyon olmuştur. Bilgisayar bilimi dersinin içeriğini tanımlamak amacıyla CAS (2012) tarafından tasarlanan öğretim programı incelenen diğer bir kaynak olmuştur. Bu kaynakta, bilgisayar biliminin temel süreçlerinde BD'ye odaklandığı savunulmuştur. Bu süreçlerin ise soyutlama (modelleme, parçalara ayırma ve genelleme) ve programlamayı içerdiği ifade edilmiştir. Daha sonra, CollegeBoard'un (2015; 2016) hazırladığı öğretim programları incelenmiştir. Bu kaynaklarda, öğretim sürecinin tasarlanması ve yürütülmesinde uygulayıcılara rehberlik etmesi, öğrencilerin bilgisayar biliminin araçları ve yöntemleriyle tanıştırılması ve BD uygulamaları ile farklı beceriler geliştirilmesi için kazanımlar yazıldığı görülmüştür. CollegeBoard (2015) kaynağında kazanımların; yaratıcılık, veri ve bilgi, soyutlama, programlama, algoritma, internet ve küresel etki olmak üzere yedi "büyük fikir" etrafında şekillendirildiği belirlenmiştir. Bu fikirler, CollegeBoard (2016) kaynağında yerini yaratıcı üretim, veriler, bilgisayar sistemleri ve ağları, algoritmalar ve programlama ve bilgi işlemin etkisine bırakmıştır.

Öğretim hedeflerin oluşturulmasında, Dorling ve Walker'in (2014) farklı eğitim seviyeleri için kazanımlar önerdiği ve bunların BD becerileri ile ilişkilerini açıkladığı çalışma bir başka kaynak olarak incelenmiştir. Dorling ve Walker (2014), bilgisayar bilimlerinin yalnızca programlamayı değil, problem çözüme süreçlerini de içerdiğini vurgulamışlardır. Ayrıca, BD'nin karmaşık problemleri analiz etme, modelleme, çözüme ve çözümü bilgisayar aracılığıyla uygulama becerilerini geliştirdiğini belirtmişlerdir. Çalışmada kazanımların; algoritma geliştirme, programlama ve ürün geliştirme başlıklarında toplandığı ve bunların bazılarının birden fazla beceriye işaret ettiği belirlenmiştir. Ayrıca, ürün geliştirme ve programlama becerilerinin tamamının algoritma tasarımı ile ilişkilendirildiği tespit edilmiştir.

Doküman analizi sonucunda; BD becerilerinin gelişiminde algoritma, programlama, ürün geliştirme (özgün proje tasarlama ve üretme), yaratıcılık, problem çözüme, parçalara ayırma, soyutlama, veri (toplama, çözümlleme ve sunma), otomasyon, paralelleştirme ve simülasyon olmak üzere 11 tema belirlenmiştir. Bu noktada; algoritma, programlama, ürün geliştirme, problem çözüme ve soyutlama en çok üzerinde durulan ve yüksek önem değerlerine sahip temalar olarak öne çıkmıştır.

## **Sonuç, Tartışma ve Öneriler**

Bu çalışmada, bilgisayar oyunu programlama aracılığıyla BD becerilerini geliştirmek ve değerlendirmek için öğretmenlere rehberlik edecek bir çerçevenin sunulması amaçlanmıştır. BD becerileri ve yaklaşımlarına yönelik öğretim etkinlikleri ve kazanımlar içeren bu çerçeve, bir durum çalışması ile oluşturulmuştur. Bu bağlamda, alanyazın incelemesi, ihtiyaç analizi, öğrenen analizi ve öğrenme hedeflerinin belirlenmesi için doküman analizi yapılmış ve sonuçlar yorumlanmıştır.

Alanyazının detaylı incelemesi ile BD'nin programlama becerilerini içerdiği ancak bunlarla sınırlı olmadığı sonucuna varılmıştır. Bu bağlamda, bilgisayar oyunu programlama ortaokul düzeyindeki öğrenciler için tasarlanan çerçevenin geliştirilmesinde bir araç olarak değerlendirilmiştir. 21. yüzyıl becerileri kapsamında birçok ülkenin öğretim programlarına alınmış olmasına rağmen, BD becerilerinin programlamayı içerip içermediği hakkındaki tartışmalar devam etmektedir (Denning, 2017). Bazı kaynaklarda BD becerilerinin gelişimi, programlama öğrenmeyle ilgili doğrudan ilişkilendirilmekte (örn. Denning, 2017; Laura-Ochoa ve Bedregal-Alpaca, 2022) ve programlama BD'nin ayrılmaz bir parçası olarak değerlendirilmektedir (örn. Grover ve Pea, 2018; Mouza, Pan, Yang ve Pollock, 2020). Ancak bu çalışma, bu görüşe alternatif bir bakış açısı sunmaktadır. Nitekim Wing (2008) de, BD'nin programlama dili okuryazarlığından daha fazlasını ifade ettiğini belirtmiştir. Ayrıca, BD'nin gelişimi için programlama becerilerinin ötesini işaret eden diğer araştırmalar da bulunmaktadır (örn. Bers, 2021; Dorling ve Walker, 2014; Kılıç, 2023). BD, problemleri analiz etme, algoritmik çözüm yolları oluşturma ve verileri anlama gibi süreçleri içermektedir ve programlamanın bunları uygulama noktasında etkili bir rolü vardır. Dahası, alanyazında BD'nin geliştirilmesi için bazı çalışmalarda metin tabanlı yaklaşımların takip edildiği görülmüştür (örn. Bai, Wang ve Zhao, 2021; Bufasi, Hoxha, Cuka ve Vrtagi, 2022; Kroustalli ve Xinogalos, 2021; Sun ve Liu, 2024; Weintrop ve Wilensky, 2019). Bu uygulamaların özellikle yazılım geliştirici olmayı hedefleyen öğrenciler için faydalı olacağı söylenebilir.

Ancak bu çalışma ile Wing'in (2006) de önerdiği gibi yazılım geliştirici olmayı hedeflemeyen öğrenciler için blok tabanlı oyun programlama aracılığıyla alternatif bir yaklaşım sunulmuştur. Benzer biçimde, programlamanın özellikle görsel programlama dilleri kullanıldığında BD'yi geliştirmenin etkili bir yolu olduğunu belirten araştırmalar da bulunmaktadır (örn. Fulop, Udvaros, Guban ve Sandor, 2022; Sun, Looi, Li, Zhu, Zhu, ve Cheng, 2024; Voogt vd., 2015; Wu, Lin, Wang ve Huang, 2023).

Bu çalışmada gerçekleştirilen taramalar sonucunda, programlama becerilerinin yanı sıra BD becerilerine (parçalarına ayırma, örüntü tanıma, soyutlama, algoritma tasarlama ve hata ayıklama) ve BD'de problem çözme yaklaşımlarına (deneyimleme, üretme, hata ayıklama, azimli olma ve işbirliği yapma) odaklanılmasının önem taşıdığı sonucuna varılmıştır. Bu bağlamda, her bir beceri ve yaklaşım için öğrenme etkinlikleri ve kazanımları içeren bir çerçeve oluşturulmuştur. Benzer biçimde, Kim, Leftwich ve Castner (2024) erken yaşta BD eğitiminin öğrenme hedeflerinin neler olabileceği konusunda, öğretim programı tasarımında aktif aktörler olarak betimledikleri öğretmenlerin görüşlerini almışlardır. Buna göre, öğretmenlerin birincil öğrenme hedeflerinden biri olarak temel BD becerilerinin geliştirilmesini önerdiklerini ve bu becerileri de algoritmalar, örüntü tanıma ve parçalara ayırma şeklinde sıraladıklarını ifade etmişlerdir. Ayrıca Mannila vd. (2023) de Finlandiya'da gerçekleştirdikleri durum çalışmasıyla müfredatlar arası bir öğretim dizisi önerisi sunmuştur. Bu öğretim dizisinde deneyimleme, üretme, hata ayıklama, işbirliği, azimli olma, algoritma ve değerlendirme merkeze alınmıştır. Bununla birlikte, Tsortanidou, Daradoumis ve Barberá, (2023) kodlama gerektirmeyen bir yaklaşımla BD becerilerine odaklanan bir öğretim programı önermiştir. Ayrıca, bu becerilerin öğrencilerin problem çözme süreçlerini yönlendirmelerine ve yaratıcılık ve işbirliği becerilerini de geliştirmelerine destek sağladığını ifade etmiştir. Alanyazında, öğrencilerin BD ile gerçek dünya problemlerine sistematik bir şekilde yaklaşmayı, çözüm yolları üretmeyi ve alternatif çözümleri değerlendirmeyi öğrenirken aynı zamanda dayanıklılık ve problem çözme azmi kazandıkları da vurgulanmaktadır (Chen, Sonnert, Sadler ve Malan, 2020). Sonuç olarak, geliştirilen öğretim programının öğrencilerin programlamanın ötesinde BD'yi anlamalarına ve uygulamalarına olanak tanıyacağı düşünülmektedir. Ayrıca, öğrencilerin gelecekteki problemlerinin çözümünde deneyimleme, azimli olma ve işbirliği yapma gibi yaklaşımları kullanabilen bireyler haline gelmelerine yardımcı olacağı öngörülmektedir.

İhtiyaç analizi sonucunda, öğretmenler tarafından ortaokul düzeyinde BD becerisinin geliştirilmesinde kullanılmak üzere bir çerçeve programın tasarlanmasına ihtiyaç olduğu ortaya çıkmıştır. Benzer biçimde Belmar (2022), birçok öğretmenin öğrenme ortamını BD gelişimi destekleyecek biçimde tasarlama konusunda deneyimi olmadığını ve bu alanda uygun öğretim modellerine ihtiyaç duyulduğunu ifade etmiştir. Ayrıca, Mannila vd. (2023) öğretmenlerin programlama ve BD becerilerinin gelişimi konusunda yeterli seviyede bilgi ve becerilerinin olmadığını ve bu bağlamda öğretim aktivitelerinin geliştirilmesi gerektiğini vurgulamıştır. BD'nin öğretim programlarına entegrasyonunun önündeki engellerin incelendiği diğer çalışmalarda ise öğrenim düzeyi ve branşı ne olursa olsun öğretmenlerin BD'yi kavrama düzeylerinin çok düşük olduğuna dikkat çekilmiştir (Kite ve Park, 2023). Bu bağlamda, öğretmenlerin BD'yi genellikle problem çözme becerisi ile ilişkilendirdikleri, ancak BD uygulamalarını teknoloji ile birleştirerek daha derinlemesine bir kavrayış geliştirmede nadiren başarılı oldukları ortaya çıkmıştır (Peracaula-Bosch, Estebanell-Minguell, Couso ve González-Martínez, 2020). Çalışmanın bir başka sonucu olarak, BD becerilerinin eğitim sistemi içerisine küçük yaşlardan itibaren dahil edilmesinin önemli olduğu, ancak bu konudaki gereksinimlerin yeterince karşılanmadığı dikkat çekmiştir. Barr ve Stephenson (2011) da, öğrencilerin dijital araçlardan güçlü bir biçimde etkilenen bir dünyada yaşadıklarını ve dolayısıyla BD becerilerini geliştirmek için üniversiteye düzeyine gelmelerini beklemenin akıllıca olmadığını ifade etmiştir. Çalışmada, öğretim programında BD becerilerine yer verilmesinin; problem çözme, programlama, BD becerileri ve BD yaklaşımlarının geliştirilmesi açısından önemli katkılar sağlayabileceği sonucuna ulaşılmıştır. Benzer biçimde, bu bağlamda önerilen öğretim programlarının öğrencilerin programlama becerilerinin (Mannila vd., 2023), BD becerilerinin (Tsortanidou vd., 2023) ve BD de problem çözme yaklaşımlarının gelişmesine (Mannila vd., 2023) katkı sağladığı görülmüştür. Bununla birlikte, tasarlanacak öğretim etkinliklerinin; öğrencilerin programlamaya yönelik olumsuz tutumlarını değiştirmeye, derse olan ilgi ve katılımlarını artırmaya, üretme ve programlama konusundaki motivasyonlarını yükseltmeye, merak duygularını canlı

tutarak kendi öğrenme süreçlerini yönetme becerilerini güçlendirmeye yönelik ve daha fazla üretmeleri için cesaretlendirilmelerine katkı sağlayacağı belirlenmiştir.

Öğrenen analizi sonucunda, BTY dersi öğretim programında BD becerilerine yönelik birçok kazanım bulunmasına rağmen, öğretmenlerin bu becerilerin geliştirilmesinde bir takım zorluklarla karşılaştıkları ortaya çıkmıştır. Bu doğrultuda, öğrencilerin tamamının daha önce programlama dersi aldığı ve en az bir programlama aracına aşına olduğu, ancak bazılarının algoritma tasarımı, değişkenler döngü ve karar yapıları ve hata ayıklama konularında eksikliklerinin olduğu ortaya çıkmıştır. Ek olarak, yapılandırmacı kurama dayalı ve öğrencinin aktif katılımını önceleyen süreç odaklı değerlendirmenin benimsendiği öğretim programında, öğretmenlerin tüm kazanımlar ve dolayısıyla dönem sonunda özgün proje üretimi için yeterli zamanı bulamadıkları ortaya çıkmıştır. Benzer biçimde, alanyazında, yetersiz teknolojik altyapı, zaman kısıtlamaları ve düşük teknolojik okuryazarlığa sahip öğrenciler gibi nedenlerle öğretmenlerin BD'yi sınıf içi uygulamalarına entegre edemediklerinin altı çizilmiştir (Kite ve Park, 2023; 2024; Lockwood ve Mooney, 2018; Searle, Tofel-Grehl, Fischback ve Hansen, 2023; Israel vd., 2015). Bu durumda, çalışmada tasarlanan etkinliklerin yeni değerlendirme araçlarının geliştirilmesindeki eksikliklerin giderilmesi için öğretmenlere destek sağlayabileceği öne sürülebilir.

Öğretim programlarının detaylı analizi, öğrencilerin geliştirmesi gereken temel becerilerin algoritma, programlama, ürün geliştirme, problem çözme, parçalara ayırma ve soyutlama olduğunu açıkça göstermiştir. Bu sonuçlar, ihtiyaç ve öğrenen analizi sonuçları da dikkate alınarak değerlendirilmiştir. Buradan hareketle, mevcut çalışmada kullanılmak üzere kazanımlar oluşturulmuştur. Bu kazanımlar, programlama becerisi ve BD becerileri başlıkları altında toplanmıştır. Tablo 3'te, kazanımlar, kazanımların ilişkilendirildiği BD becerileri ve araştırmacılar tarafından oluşturulan kazanımlara ait kodlar sunulmuştur.

Tablo 3

*Çalışma Kapsamında Geliştirilen Kazanımlar*

Kazanımlar	BD	Kod
<b>Programlama Becerisi</b>		
Görsel programlama aracını ve özelliklerini keşfeder.	-	BD1
Görsel programlama aracında verilen bir oyunun işlevlerini anlar.	-	BD2
Programların kesin talimatları izleyerek çalıştığını anlar.	AT	BD3
Verilen hedeflere ulaşmak için algoritmalar tasarlayarak programlar oluşturur.	AT	BD4
Doğrusal mantık, döngüler, karar ve çoklu karar yapıları içeren programlar oluşturur.	AT	BD5
Programlardaki basit anlamsal hataları algılar ve düzeltir.	HA	BD6
<b>Bilgisaymsal Düşünme Becerileri</b>		
Verilen karmaşık bir problemi alt parçalarına ayırır.	PA	BD7
Farklı problemlerin çözümünde kullanılacak genelleştirilebilir örüntüleri belirler.	ÖG	BD8
Farklı problemlerin çözümündeki örüntüleri tanımlamak için modeller oluşturur.	SY	BD9
Problemin çözümü için geliştirdiği algoritmayı yapılandırılmış bir gösterim kullanarak sunar.	AT	BD10
Bir görsel programlama dili kullanarak özgün bir ürün tasarlar ve yinelemeli olarak geliştirir.	AT, PA, ÖG, SY, HA	BD11
Karmaşık problemlerin her bir alt parçası için çözümler oluşturur ve bunları uygular.	AT, PA	BD12
Benzer problemlere özyinelemeli olarak geliştirilen bir çözümü anlar, aynı çözümü problemin daha küçük örneklerine tekrar tekrar uygulayarak geneller.	AT, ÖG	BD13
Benzer problemlerin çözümü için geliştirdiği algoritma ve modellerin etkililiğini değerlendirir ve bunları uygular.	AT, SY, ÖG	BD14
Problemlerin çözümü için geliştirdiği algoritmaları bir görsel programlama dili kullanarak oluşturur.	AT	BD15
Hatalı bir algoritmadaki hataları tespit eder ve bunları düzeltir.	HA	BD16
Algoritmalar için performans kavramını anlar ve bazı algoritmaların aynı görev için farklı performans özelliklerine sahip olduğunu anlar.	AT, HA	BD17

Oluşturulan kazanımlar ile birlikte, hedef kitleye yönelik bilgisayar oyunu programlama etkinlikleri geliştirilmiştir. Ayrıca, öğrenmeyi kolaylaştırmayı hedefleyen ve değerlendirme aracı olarak kullanılacak çeşitli formlar ve çalışma yapıları tasarlanmıştır (Özmen, 2020). Tablo 4'te, geliştirilen içerik, içeriğin ilişkili olduğu BD yaklaşımları ile çalışma kapsamında oluşturulan ve BD becerilerini içeren kazanım karşılıkları sunulmuştur. İlgili etkinliklerin değerlendirilmesinde kullanılacak söz konusu formlar ve çalışma yapıları parantez içinde belirtilmiştir.

Tablo 4

## Oyun Programlama Aracılığıyla BD Becerilerinin Geliştirilmesine Yönelik Öğretim Etkinlikleri İçeriği

Yaklaşım	İçerik	Kazanım
DENEYİMLEME	<b>Programlama aracını keşfetme:</b> Görsel programlama aracının öğrenciler tarafından keşfedilebilmesi için aracın tanıtımı yapılarak ilk deneyimlerin tartışılması ve paylaşılması teşvik edilir. <b>(İŞBİRLİĞİ YAPMA)</b>	BD1
	<b>Mevcut oyunları analiz etme:</b> Oyun programlama aracında mevcut olan örnek oyunlar, tasarım elemanlarına, oyun mekaniklerine, oyun elementlerine ve kodların bir oyun içindeki kullanımına yönelik fikir edinilmesi için oynatılır.	BD2
	<i>Oyun mekanikleri, oyunun hikayesi ve amacıyla bağlantılı olarak oyun içinde sürekli tekrarlanan eylemlerdir (hareket etme, toplama, çarpma, yarışma, zıplama, kaçma, keşif yapma, hedefleme vs.). Oyun elementleri ise oyunun çeşitli yerlerinde kullanılarak oyun deneyimini zenginleştiren ve oyuncuyu oyuna bağlayan unsurlardır (skor, ödüller, süre, seviyeler, puan, zamanla yarışma, geri bildirim vs.)</i>	
	<b>Oyun geliştirme:</b> Oyun geliştirme bilgi ve becerilerinin kazandırılması için seçilen aracın özelliklerine göre şekillendirilebilecek çeşitli oyun programlama etkinlikleri ile (nesnel arası etkileşim, skor, rekabet, süre tutma, hedefe ulaşma, engelleri aşma, yarışma, değişkenler vs.) örnek oyunlar geliştirilir. Her bir oyunda; tasarım araçlarının kullanım biçimine, programlama yapılarına, oyun elementlerine ve mekaniklerine ilişkin basamak basamak bilgi verilir.	BD3 BD4 BD5 BD6
	<i>Bu aşamada, programlama süreci modellenerek her bir öğrencinin bireysel olarak tekrarlaması sağlanır. Daha sonra öğrencinin hazırladığı oyunu kaydetmesine ve denemesine yardımcı olunur.</i>	
	<b>Özgün projeye hazırlık:</b>	
AZIMLI OLMA	1. İlk karakterin tasarımı yaptırılır (Kendini Tanıtma Formu).	BD3
	2. Basit bir oyun hikayesi yazdırılır, çizimi yaptırılır (Çizimli Oyun Hikaye Formu).	BD4
	3. Hikaye, oyun programlama aracı yardımıyla kodlatılır.	BD6
	<i>Örnek karakterlerin tasarlanması ve oyunların geliştirilmesi, özgün proje üretimi sürecinin prova edilerek kolaylaştırılmasına yardımcı olmaktadır.</i>	
	<b>Hikaye oluşturma:</b> Üretilecek oyunların hikayeleri yazdırılır (Oyun Hikaye Formu). Öncesinde, hikaye yazımına ilişkin kurallar verilir (karakter sayısı, tasarım elemanlarının özellikleri vs.). Örnek hikayeler, sınıfta okutulur ve tartışılması sağlanır. Tüm hikayeler; kurallar, oyun elementlerini içermeye, görsel bir programlama aracı ile tasarlanabilir olma açısından değerlendirilir ve dönütler verilir. Hikayelerin verilen dönütler temelinde gözden geçirilerek düzeltilmesi istenir.	BD3 BD4 BD6
	<b>Parçalara ayırma:</b> Hikayenin, alt parçalarına (karakterler, zaman, mekan vs.) ayrılması istenir. Bir başka deyişle, karmaşık problemin (oyun) daha kolay çözülebilir alt parçalara ayrıştırılması beklenir (Parçalara Ayırma Çalışma Yaprağı).	BD7
	<b>Örüntü tanıma:</b> Hikayelerdeki benzer parçaların listelenmesi istenir (Örüntü Tanıma Çalışma Yaprağı). Probleme ve problemin alt parçalarına ilişkin örüntülerin ortaya konulması beklenir.	BD8
	<b>Soyutlama:</b> Örüntülere dayalı olarak kendi oyunlarının diğerlerinininkinden nasıl farklılaştığını fark etmeleri ve oyununu gereksiz detaylardan arındırmaları istenir (Soyutlama Çalışma Yaprağı).	BD9
	<b>Algoritma tasarımı:</b> Hikayesi yazılan oyun için algoritma oluşturulması istenir. Hikayelerin oyun programlama aracında tam, doğru ve anlaşılır bir biçimde tasarlanabilmesi amacıyla uygun basamaklara ayrılması istenir (Hikaye Tahtası).	BD10
	<b>Özgün proje:</b> Hikayesi daha önce gözden geçirilerek düzenlenmiş olan son oyun, programlama aracı ile tasarlanır ve kodlanır. Hikayelerin oyunlara dönüştürülerek özgün ürünlerin geliştirilmesi, ürünlerin akranlarla paylaşılması, değerlendirilmesi, iyileştirilmesi ve ürün geliştirme sürecine ilişkin deneyimlerin yansıtılması beklenir.	BD11 BD12 BD13 BD14 BD15 BD16
<b>(İŞBİRLİĞİ YAPMA)</b>		
HATA AYIKLAMA	<b>Hata ayıklama:</b> Oyundaki hataların bulunarak düzeltilmesi için önceden hazırlanıp kodları değiştirilmiş bir oyun tüm öğrencilere dağıtılır. Hataların, Berry (2015, s.29) tarafından önerilen dört hata ayıklama basamağına uygun şekilde ifade edilmesi ve oyun üzerinde düzeltilmesi istenir (Hata Ayıklama Çalışma Yaprağı). Bu basamaklar aşağıda sıralanmıştır.	BD16 BD17
	1. Ne olması gerekiyor, tahmin et.	
	2. Ne olduğunu bul.	
	3. Hatayı nasıl düzeltebilirsin?	
	4. Hatayı düzelt.	

Sonuç olarak, ortaokul öğrencilerinin oyun programlama aracılığıyla BD becerilerinin geliştirilmesi ve değerlendirilmesi için bir çerçeve tasarlanmıştır. Bu bağlamda, alanyazından hareketle, BD'nin farklı disiplinlerde karmaşık problemlerin çözümünde kullanılabilecek bir beceri seti olduğu sonucuna ulaşılmıştır. Bu noktada, BD becerileri olarak parçalara ayırma, örüntü tanıma, soyutlama, algoritma tasarımı ve hata ayıklamaya odaklanıldığını görülmüştür. Bu beceriler, BD'de problem çözme yaklaşımları olarak deneyimleme, üretme, hata ayıklama, azimli olma ve işbirliği yapma ile ilişkilendirilmiştir. Ayrıca, programlamanın BD becerilerinin kazandırılmasında bir araç olarak kullanılabileceği, bununla birlikte, BD'nin programlama dili okuryazarlığından daha fazlasını ifade ettiği belirlenmiştir. Bu perspektiften bakıldığında BD, problemlerin tanımlanması ve çözülmesinde yenilikçi yaklaşımları benimseyerek çeşitli stratejiler geliştiren ve bunları etkili bir şekilde kullanabilen bireyler için temel bir yeterlilik olarak değerlendirilmiştir. Bu bağlamda, bazı ülkelerin bilgisayar

bilimini ve BD'yi öğretim programlarına dahil ettikleri ve BD becerilerine odaklandıkları dikkat çekmiştir. Çalışmanın diğer sonuçları, erken yaşta BD becerilerinin geliştirilmesinin, programlama ve problem çözme becerilerini desteklemesinin yanısıra duyuşsal ve bilişsel yönden birçok kazanım sağlayacağını göstermiştir. Ancak öğretmenlerin BD'yi sınıf içi uygulamalara entegre etme konusunda yeterli düzeyde bilgi ve anlayışa sahip olmama, zaman kısıtlamaları, düşük teknolojik okuryazarlığa sahip öğrenciler gibi birçok engelle karşılaştıkları tespit edilmiştir. Bu bağlamda tasarlanan çerçevenin, öğretmenlere BD'nin geliştirilmesi ve değerlendirilmesi konusunda destek sağlayacak bir kaynak olduğu düşünülmektedir. Bununla birlikte, ortaokul düzeyinde öğrenim gören öğrenciler için tasarlanan bu çerçevenin ulusal öğretim programlarına da katkı sağlaması beklenmektedir.

Bu makalede geliştirilen çerçevenin gerçek dünya öğretim ortamlarında uygulanması ve öğrenci performansına etkilerinin incelenmesi çerçevenin etkililiğinin değerlendirilmesine yardımcı olabilir. Çalışmada, ortaokul düzeyine odaklanılmıştır. Gelecekteki araştırmalar, farklı yaş grupları veya eğitim seviyelerindeki öğrencilerin ihtiyaçlarına yönelik olarak yeni çerçeve ve modeller üretilmesini içerebilir. Öğretim etkinliklerinin başarıyla yürütülmesinde, öğretmenlerin programlama becerileri ve bilgisayar oyunu programlama konusundaki bilgi düzeyleri kritiktir. Öğretmenler, öğretim programını öğrencilerin gereksinimlerine uygun bir şekilde uygulamak için gereken bilgiye ve becerilere sahip olmalıdır. Diğer yandan, geliştirilen çerçevenin etkililiğini değerlendirmek için çeşitli değerlendirme stratejileri belirlenmelidir. Hem süreç değerlendirmesi (örneğin, proje bazlı çalışmalar) hem de ürün değerlendirmesi (örneğin, üretilen oyunların kalitesi) içeren değerlendirme yöntemleri kullanılabilir.

### Araştırmacıların Katkı Oranı

Bu araştırmaya ilk yazar %70, ikinci yazar ise %30 oranında katkıda bulunmuştur.

### Çıkar Çatışması

Bu araştırmada herhangi bir çıkar çatışması söz konusu değildir.

**Teşekkür ve Açıklama:** Bu çalışma, ikinci yazarın danışmanlığını yürüttüğü birinci yazarın doktora tezinden üretilmiştir. Ayrıca, birinci yazara Bilim İnsanı Destekleme Programı, 2211/E Doğrudan Yurt İçi Doktora Bursu ile TÜBİTAK tarafından destek sağlanmıştır (Başvuru No: 1649B031300059).

### Kaynaklar

- Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., ... Zagani, J. (2016). A K-6 computational thinking curriculum framework: Implication for teacher knowledge. *Educational Technology & Society*, 19(3), 47–57.
- Australian Curriculum, Assessment and Reporting Authority. (2015). *The shape of the Australian curriculum*. Australian Government Publishing Service. Canberra. [https://docs.acara.edu.au/resources/The\\_Shape\\_of\\_the\\_Australian\\_Curriculum\\_V3.pdf](https://docs.acara.edu.au/resources/The_Shape_of_the_Australian_Curriculum_V3.pdf)
- Bai, H., Wang, X., & Zhao, L. (2021). Effects of the problem-oriented learning model on middle school students' computational thinking skills in a python course. *Frontiers in Psychology*, 12, 771221.
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48–54.
- Belmar, H. (2022). Review on the teaching of programming and computational thinking in the world. *Frontiers in Computer Science*, 4, 997222.
- Berland, M., & Lee, V. R. (2011). Collaborative strategic board games as a site for distributed computational thinking. *International Journal of Game-Based Learning*, 1(2), 65–81.

- Berry, M. (2015). *QuickStart Primary Handbook: A CPD toolkit for primary teachers*. British Computer Society.
- Bers, M. U. (2021). From computational thinking to computational doing. In *Teaching computational thinking and coding to young children* (pp. 1–20). IGI Global.
- Blikstein, P., Worsley, M., Piech, C., Sahami, M., Cooper, S., & Koller, D. (2014). Programming pluralism: Using learning analytics to detect patterns in the learning of computer programming. *Journal of the Learning Sciences*, 23(4), 561–599.
- Bowen, G. A. (2009). Document analysis as a qualitative research method. *Qualitative Research Journal*, 9(2), 27–40.
- Bufasi, E., Hoxha, M., Cuka, K., & Vrtagic, S. (2022). Developing student's comprehensive knowledge of physics concepts by using computational thinking activities: Effects of a 6-week intervention. *International Journal of Emerging Technologies in Learning*, 17(18).
- Chen, C., Sonnert, G., Sadler, P. M., & Malan, D. J. (2020). Computational thinking and assignment resubmission predict persistence in a computer science MOOC. *Journal of Computer Assisted Learning*, 36(5), 581–594.
- CollegeBoard (2015). *AP Computer science: Principles. Course planning and pacing guide*. <https://apcentral.collegeboard.org/media/pdf/ap-comp-sci-principles-cppg-kick.pdf>.
- CollegeBoard (2016). *AP Computer science: Principles. Course and exam description*. <https://secure-media.collegeboard.org/digitalServices/pdf/ap/ap-computer-science-principles-course-and-exam-description.pdf>.
- Computer Science Teacher Association. (2017). CSTA K-12 computer science standards, revised 2017. <http://www.csteachers.org/standards>.
- Computing at School. (2012). *Computer science: A curriculum for schools*. <http://www.computingatschool.org.uk/data/uploads/ComputingCurric.pdf>.
- Computing at School. (2014). *Computational thinking*. CAS Barefoot. <https://www.computingatschool.org.uk/media/kscbloob/computationalthinking.pdf>.
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., ... Woollard, J. (2015). *Computational thinking. A guide for teachers*. Computing at School. Charlotte BCS. The Chartered Institute for IT.
- Csizmadia, A., Standl, B., & Waite, J. (2019). Integrating the constructionist learning theory with computational thinking classroom activities. *Informatics in Education*, 18(1), 41-67.
- Curzon, P., Dorling, M., Ng, T., Selby, C., & Woollard, J. (2014). *Developing computational thinking in the classroom: A framework*. Swindon, GB. Computing at School.
- Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, 60(6), 33–39.
- Department for Education (2013). *Computing programmes of study: Key stages 1 and 2. National curriculum in England*. <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study>.
- Dorling, M., & Walker, M. (2014). *Computing progression pathways with Computational thinking*. Computing at School. <http://community.computingatschool.org.uk/resources/2324>.
- Finnish National Board of Education (2016). *Curriculum in Finland*. [https://www.dge.mec.pt/sites/default/files/Noticias\\_Imagens/1\\_curriculum\\_in\\_finland.pdf](https://www.dge.mec.pt/sites/default/files/Noticias_Imagens/1_curriculum_in_finland.pdf).

- Fulop, M. T., Udvaros, J., Guban, A., & Sandor, A. (2022). Development of computational thinking using microcontrollers integrated into OOP (Object-Oriented Programming). *Sustainability*, 14(12), 7218.
- Futschek, G. (2006). Algorithmic thinking: The key for understanding computer science. *International Conference on Informatics in Secondary Schools-Evolution and Perspectives*, (pp. 159–168). Springer.
- Grover, S., & Pea, R. (2018). *Computational Thinking: A competency whose time has come*. In S. I. E. Barendsen, & C. Shulte (Eds.), *Computer science education: Perspectives on teaching and learning in school* (pp. 19-37). Bloomsbury Academic.
- Herrero-Alvarez, R., Miranda, G., Leon, C., & Segredo, E. (2022). Engaging Primary and Secondary School Students in Computer Science through Computational Thinking Training. *IEEE Transactions on Emerging Topics in Computing*, 11(1), 56-69.
- Hunsaker, E. (2018). *Understanding computational thinking*. Brigham Young University, Provo.
- International Society for Technology in Education. (2016). ISTE standards for students. ISTE. <https://iste.org/standards/students>
- International Society for Technology in Education. (2017). ISTE standards for educators. <https://iste.org/standards/educators>.
- Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education*, 82(2015), 263–279.
- Kılıç, S. (2023). Background of the Relationship between Programming and Computational Thinking. In *Innovative Digital Practices and Globalization in Higher Education* (pp. 203–224). IGI.
- Kim, C., Belland, B. R., Baabdullah, A., Lee, E., Dinç, E., & Zhang, A. Y. (2021). An ethnomethodological study of abductive reasoning while tinkering. *AERA Open*, 7.
- Kim, J., Leftwich, A., & Castner, D. (2024). Beyond teaching computational thinking: Exploring kindergarten teachers' computational thinking and computer science curriculum design considerations. *Education and Information Technologies*, 1-37.
- Kite, V., & Park, S. (2023). What's computational thinking?: Secondary science teachers' conceptualizations of computational thinking (CT) and perceived barriers to CT integration. *Journal of Science Teacher Education*, 34(4), 391-414.
- Kite, V., & Park, S. (2024). Context matters: Secondary science teachers' integration of process-based, unplugged computational thinking into science curriculum. *Journal of Research in Science Teaching*, 61(1), 203-227.
- Krippendorff, K. (2004). Reliability in content analysis: Some common misconceptions and recommendations. *Human Communication Research*, 30, 411–433.
- Kroustalli, C., & Xinogalos, S. (2021). Studying the effects of teaching programming to lower secondary school students with a serious game: A case study with Python and CodeCombat. *Education and Information Technologies*, 26(5), 6069-6095.
- Laura-Ochoa, L., & Bedregal-Alpaca, N. (2022). Incorporation of computational thinking practices to enhance learning in a programming course. *International Journal of Advanced Computer Science and Applications*, 13(2).
- Lincoln, Y. S., & Guba, E. G. (1985). *Naturalistic inquiry*. Sage.



- Liu, Z., Zhi, R., Hicks, A. & Barnes, T., (2017). Understanding problem solving behavior of 6-8 graders in a debugging game. *Computer Science Education*, 27(1), 1–29.
- Lockwood, J., & Mooney, A. (2018). Computational thinking in education: Where does it fit? A systematic literary review. *International Journal of Computer Sciences and Engineering Systems*, 2(1), 41–60.
- Lodi, M., & Martini, S. (2021). Computational thinking, between Papert and Wing. *Science & Education*, 30(4), 883–908.
- Mannila, L., Leinonen, T., Bauters, M., & Veermans, M. (2023). Student and teacher co-agency when combining CT with arts and design in a cross-curricular project. *Computers and Education Open*, 4, 100132.
- Merriam, S. B. (2009). *Qualitative research: A guide to design and implementation* (3rd ed.). CA: Jossey-Bass.
- Miles, M., & Huberman, A. (1994). *Qualitative data analysis: An expanded sourcebook* (2nd ed.). Sage.
- Milli Eğitim Bakanlığı (2018). *Bilişim teknolojileri ve yazılım dersi öğretim programı (Ortaokul 5. ve 6. sınıflar)*. <https://mufredat.meb.gov.tr/ProgramDetay.aspx?PID=374>.
- Mouza, C., Pan, Y. C., Yang, H., & Pollock, L. (2020). A multiyear investigation of student computational thinking concepts, practices, and perspectives in an after-school computing program. *Journal of Educational Computing Research*, 58(5), 1029-1056.
- New Zealand Ministry of Education (2017). *The ministry of education annual report 2017*. <https://www.education.govt.nz/assets/Documents/Ministry/Publications/Annual-Reports/2017-MOE-Annual-Report-web.pdf>.
- Özmen, B. (2020). *Programlama öğretiminde bilgisayarlı düşünme becerilerinin geliştirilmesine yönelik oyun tabanlı bir tasarım modeli önerisi [Yayımlanmamış Doktora Tezi]*. Hacettepe Üniversitesi. <https://openaccess.hacettepe.edu.tr/xmlui/handle/11655/23259>
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.
- Patton, M. Q. (2014). *Qualitative research and evaluation methods: Integrating theory and practice* (4th ed.). Sage.
- Pelanek, R., & Effenberger, T. (2023). The landscape of computational thinking problems for practice and assessment. *ACM Transactions on Computing Education*, 23(2), 1–29.
- Peracaula-Bosch, M., Estebanell-Minguell, M., Couso, D., & González-Martínez, J. (2020). What do pre-service teachers know about computational thinking? *Aloma: Revista de Psicologia, Ciències de l'Educació I de l'Esport*, 38(1), 75–86.
- Rich, K. M., Spaepen, E., Strickland, C., & Moran, C. (2019). Synergies and differences in mathematical and computational thinking: Implications for integrated instruction. *Interactive Learning Environments*, 28(3), 272–283.
- Searle, K. A., Tofel-Grehl, C., Fischback, L., & Hansen, T. (2023). Affordances and limitations of teachers instructional styles when teaching computer science and computational thinking. *Computer Science Education*, 33(1), 139-161.
- Selby, C. C., & Woollard, J. (2013). *Computational thinking: The developing definition*. University of Southampton.
- Strauss, A., & Corbin, J. (1998). *Basics of qualitative research techniques*. Sage.

- Sun, D., Looi, C. K., Li, Y., Zhu, C., Zhu, C., & Cheng, M. (2024). Block-based versus text-based programming: A comparison of learners' programming behaviors, computational thinking skills and attitudes toward programming. *Educational Technology Research and Development*, 72(2).
- Sun, L., & Liu, J. (2024). Effects of gamified python programming on primary school students' computational thinking skills: A differential analysis of gender. *Journal of Educational Computing Research*, 62(3), 846-874.
- Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, 148, 103798.
- Tsortanidou, X., Daradoumis, T., & Barberá, E. (2023). A K-6 computational thinking curricular framework: Pedagogical implications for teaching practice. *Interactive Learning Environments*, 31(8), 4903-4923.
- Tuhkala, A., Wagner, M. L., Iversen, O. S., & Kärkkäinen, T. (2019). Technology comprehension-combining computing, design, and societal reflection as a national subject. *International Journal of Child-Computer Interaction*, 20, 54-63.
- Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4), 715-728.
- Waite, J. L., Curzon, P., Marsh, W., Sentance, S., & Hadwen-Bennett, A. (2018). Abstraction in action: K-5 teachers' uses of levels of abstraction, particularly the design level, in teaching programming. *International Journal of Computer Science Education in Schools*, 2(1), 14-40.
- Wahyuni, D. S., Rozimela, Y., Ardi, H., Mukhaiyar, M., & Darmansyah, D. (2022). PROSPER (Project, Sustainability, and Perseverance) learning model in English for computer science. *Sustainability*, 14(24), 16749.
- Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., ... Sysło, M. M. (2017). Computer science in K-12 school curricula of the 21st century: *Why, what and when?* *Education and Information Technologies*, 22, 445-468.
- Weintrop, D., & Wilensky, U. (2019). Transitioning from introductory block-based and text-based environments to professional programming languages in high school computer science classrooms. *Computers & Education*, 142, 103646.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wing, J. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical Physical and Engineering Sciences*, 366(1881), 3717-3725.
- Wing, J. M. (2011). Computational thinking - What and why? *The Link Magazine*, Spring.
- Wing, J. M. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, 25(2), 7-14.
- Woollard, J. (2016). CT Driving Computing Curriculum in England. *CSTA Voice*, 12(1), 4-5.
- Wu, T. T., Lin, C. J., Wang, S. C., & Huang, Y. M. (2023). Tracking visual programming language-based learning progress for computational thinking education. *Sustainability*, 15(3), 1983.
- Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education*, 14(1), 1-16.
- Yin, R. K. (2014). *Case study research: Design and methods* (5th ed.). Sage.

## **Extended Abstract**

### **Introduction**

Computational thinking (CT) is considered an essential 21st-century skill for solving complex problems (Barr & Stephenson 2011; Grover & Pea, 2018). According to Wing (2006), CT involves integrating the basic principles and approaches of computer science into the problem-solving process. However, Wing (2008) argued that CT focuses on how humans solve problems rather than imitating the way computers think. CT has become a crucial skill set for logical and procedural problem-solving across all disciplines, not just computer science (CSTA, 2017; ISTE, 2017).

Computer science is widely recognized as a fundamental field for developing CT skills (Voogt et al., 2015). It is noteworthy that different countries have added programming activities to their curricula to include CT in their education systems (e.g. Belmar, 2022; DE, 2013; FNBE, 2016; MEB, 2018; NZME, 2017; Tuhkala et al., 2019; Webb et al., 2017). However, some studies only consider CT in the context of programming (Denning, 2017). Therefore, providing a descriptive definition of CT and assessment methods can be challenging (Hambruch et al., 2009; Lee et al., 2011). CT encompasses problem-solving skills and an understanding of the basic principles of computer science, in addition to programming language literacy (Wing, 2006; 2008). CT aims to develop common basic problem-solving skills and identify new problems across all disciplines (Barr & Stephenson, 2011). These skills include decomposition, pattern recognition, abstraction, algorithm design, and evaluation (debugging) (Angeli et al., 2016; Csizmadia et al., 2015).

There is a need for studies that evaluate CT skills beyond programming literacy (Pelanek & Effenberger, 2023). However, it is also noteworthy that there is a lack of guidance for practitioners on how to develop teaching activities that align with the current curriculum and how to implement these activities in teaching environments. This study aims to create a framework for developing and evaluating middle school students' CT skills, as well as programming skills through computer game programming activities.

### **Method**

This study employed a case study approach. A literature review was conducted to identify the CT skills that would serve as the basis for the study. A needs analysis was carried out to define the requirements and potential educational outcomes of the planned training activities. Furthermore, a learner analysis was conducted to assess the current situation of the target group and the suitability of the teaching activities for them. Document analysis was conducted to determine the level and form of CT skills in the existing curricula, as well as the learning outcomes related to CT that are included.

Participants for the study were selected using criterion sampling, consisting of four teachers with at least five years of experience in the Information Technology field and working in the center of a province in eastern Turkey. Data for the needs and learner analysis were collected using semi-structured interview forms. To gather information on the development of CT skills at the middle school level, various documents such as curricula from Turkey and other countries, guidelines, reports, and scientific studies were utilized. The data was analyzed using content analysis techniques. The interview data was analyzed using Miles and Huberman's (1994) inductive approach, which involves data collection, summarization, visualization, inference, and confirmation. The process of document analysis followed Merriam's (2009) recommended steps, including finding appropriate documents, verifying their authenticity, creating a coding and cataloging system, and conducting data analysis. Data triangulation was used to ensure credibility and confirmability. Furthermore, expert opinions were incorporated in the development of data collection instruments. To ensure transferability, maximum diversity, and detailed description strategies were adopted. To ensure consistency, data triangulation and moderation techniques were employed. The results were found to be mutually supportive and consistent. The research received ethical approval and permission from the relevant institutions.

## **Findings**

After a thorough review of the literature, it was concluded that CT involves more than just programming skills. In addition to programming, CT includes skills such as decomposition, pattern recognition, abstraction, algorithm design, and debugging. It is important to note that these skills are interrelated and contribute to a comprehensive understanding of CT.

The needs analysis revealed seven themes: programming skills, CT skills, CT approaches, problem-solving skills, psychological outcomes, cognitive outcomes, and curriculum. It was found that teachers believed the planned teaching activities would sufficiently address students' needs for tinkering, creating, and collaborating. The results indicated that debugging and algorithm design skills were particularly crucial for CT skills, while decision structures and linear logic themes were emphasized for programming skills. These findings suggest that teachers consider CT to be equivalent to programming skills and that their understanding of CT skills may be limited. Moreover, the study's framework was perceived by teachers as having positive psychological and cognitive effects on students. It is necessary to allocate more time to the development of CT skills in the curriculum, especially through the creation of an original project, and to establish evaluation methods that can guide practitioners. Furthermore, it is crucial to create learning environments and content that cater to the needs of students.

Learner analysis revealed that the students were mostly deficient in variables, decision structures, and loop structures in terms of programming skills. However, deficiencies in debugging and algorithm design skills were observed in terms of CT skills. Additionally, it was found that most of the students had prior experience with a visual programming tool. Teachers felt that there was insufficient time or resources to undertake a product-oriented study.

The document analysis identified 11 themes related to the development of CT skills: algorithm design, programming, creating (designing and producing original projects), creativity, problem-solving, decomposition, abstraction, data (collecting, analyzing, and presenting), automation, parallelization, and simulation. At this point, the most emphasized themes were algorithm design, programming, creating, problem-solving, and abstraction.

## **Conclusion, Discussion and Recommendations**

The results of the study showed that there is a need to develop a framework for the development of CT skills and problem solving approaches in CT through computer game programming activities at middle school level. In this context, firstly, learning outcomes for programming and CT skills were developed to be used in the current study. Then, for each skill (decomposition, pattern recognition, abstraction, algorithm design, and debugging) and approach (tinkering, creating, debugging, persevering, and collaborating), a framework of game programming activities and corresponding outcomes was developed. Similarly, Belmar (2022) found that many teachers have no prior experience in designing the learning environment to support CT development and that appropriate teaching models are needed in this area. In addition, Mannila et al. (2023) developed an instructional sequence centred on experimentation, generation, debugging, collaboration, perseverance, algorithms and evaluation. As a result, a framework has been developed to address various computer game programming activities, that are appropriate to the characteristics and needs of the target audience. The primary aim of this framework is to improve students' understanding of CT beyond programming tasks and to assist practitioners in developing and assessing CT skills. Furthermore, this framework, designed for middle school students, is expected to be a valuable asset in the development of national curricula.

Future research could include the development of new frameworks and models to meet the needs of students at different ages or levels of education. It is also important to identify strategies for evaluating the effectiveness of the curricula. Additionally, evaluation methods that include both process and product evaluation can be used.