**PROGRAMLAMA SÜRECİNDE GERÇEKLİĞİN ARTTIRILMASI İÇİN BİR YAKLAŞIM**

**Merve AYDIN,** **0000-0002-4192-4504, merveaydin@trabzon.edu.tr**

**Ünal ÇAKIROĞLU,** [**0000-0001-8030-3869**](https://orcid.org/0000-0001-8030-3869)**, cakiroglu@trabzon.edu.tr**

**1** **Trabzon Üniversitesi, 2 Trabzon Üniversitesi,**

**Özet**

Programlamanın doğası ve AG teknolojisinin programlama öğrenme sürecindeki potansiyeli dikkate alındığında, programlama öğretimini kolaylaştırılması için AG’yi devreye sokacak senaryolara ihtiyaç olduğu düşünülebilir. Özellikle küçük yaştaki öğrencilerin bilgiyi en iyi şekilde şemalaştırmasını sağlamak için programlama temel konularının uygun sahneler halinde ifade edilmesi yani sürecin senaryolaştırılması önerilir. Bu çalışmada programlama yapıları ve programlama stratejilerinin öğrenilmesinin kolaylaştırılması için AG, gerçek dünyayı (programlama ortamını) sanal dünyalarla (senaryo çerçevesinde oluşturulan sanal ortam) birleştirmek, etkileşim sağlamak ve üç boyutlu (3B) nesneler sunmak amacıyla işe koşulmuştur. Bu çalışmada senaryolar hazırlanırken, senaryoda MEB (2018) belirttiği ortaokul öğrencilerine yönelik programlama kazanımları dikkate alınmıştır. Ayrıca senaryo hazırlanırken senaryo içerisinde yer alan her bir sahnenin, programlama temel konularının temel mantığını yansıtacak şekilde tasarlanmasına dikkat edilmiştir. Uzmanlarla yapılan görüşmeler sonucu öncelikle senaryolar tasarlanmış ve daha sonra her bir programlama temel yapısı için ayrı ayrı sahneler oluşturulmuştur. Programlamanın temel yapıları ve bu yapıların yürütülme mantığı temel alınarak geliştirilen senaryolar için programlama yapısı, senaryo bileşenleri ve AG teknolojisi ilişkisini ortaya koyan ortaya koyan bir akış, gerçek yaşam temelli, esnek ve ilişkili olma şeklinde bir çerçeve önerilmiştir. Önerilen senaryoların geliştirilme ve uygulama aşamalarıyla programlama çalışmalarına özellikle programlama bilgisini bilgiyi transfer etme bağlamında katkı sağlayacağı düşünülmektedir.

**Anahtar Kelimeler: programlama öğretimi, artırılmış gerçeklik, senaryo, programlama**

**Abstract**

Considering the nature of programming and the potential of AR technology in the programming learning process, it can be thought that there is a need for scenarios that will enable AR to facilitate programming teaching. In order to ensure that especially young students can schematize the information in the best way, it is recommended to express the basic programming topics in appropriate scenes, that is, to script the process. In this study, AR was employed to combine the real world (programming environment) with virtual worlds (the virtual environment created within the framework of the scenario), to provide interaction and to present three-dimensional (3D) objects in order to facilitate the learning of programming structures and programming strategies. While preparing the scenarios in this study, the programming achievements for secondary school students stated by the Ministry of National Education (2018) in the scenario were taken into account. In addition, while preparing the scenario, attention was paid to design each scene in the scenario in a way that reflects the basic logic of programming basic topics. As a result of interviews with experts, first scenarios were designed and then separate scenes were created for each programming basic structure. For the scenarios developed on the basis of the basic structures of programming and the execution logic of these structures, a framework that reveals the relationship between the programming structure, scenario components and AR technology, a flow that reveals, real-life-based, flexible and related. It is thought that the proposed scenarios will contribute to programming studies with the development and implementation stages, especially in the context of transferring programming knowledge.

Key Words: programming teaching, augmented reality, scenarios,programming

**Giriş**

Programlama öğretimi son yıllarda giderek yaygınlaşmakta ve ülkelerin okul öğretim programlarına girmektedir. Programlama çoğunlukla yazılım ile donanım arasında bir köprü olarak değerlendirilir ve bireyde problem çözme, mantıksal, analitik ve matematiksel düşünme, mantık yürütme, muhakeme, karar verme gibi birtakım üst düzey düşünme becerileri kullanmayı gerektirir. Birçok çalışma bilgisayarın yapılarına ve işlemlerine aşina olmayan kişiler için, programlamayı karmaşık ve soyut bir süreç olarak algılar. Programlama doğasındaki farklı bilgi türlerini bir arada kullanma durumundan kaynaklı karmaşıklık dışında; programlama dili yazım kurallarına uymada, algoritma oluşturmada, programlama yapılarındaki basit kavramları anlamada ve kendilerinin yazdıkları bir programdaki hataları bulmada yaşanan zorluklar literatürde sıklıkla ortaya konulmaktadır. Diğer yandan programlamaya yönelik motivasyon eksikliği de önemli faktörlerdendir. Bu zorlukların önemli nedenlerinden birisi olarak dinamik ve soyut kavramların statik materyallerle öğretilmesi olarak gösterilmektedir. Bu zorlukları aşmak ve öğrenenlere programla eğitimini daha çekici hale getirmek için birçok yollar önerilmiştir. Bu çalışmalarda çoğunlukla programlama öğretimi için ortam tasarımlarına odaklanılmış ve görsel araçlardan yararlanılmış böylelikle öğrencilerin bilgiyi kalıcı hale getirmesi ve öğrenenin ilgisini çekerek öğrenciyi, aktif kılmak amaçlanmıştır. Metin tabanlı programlama yerine blok tabanlı programların kullanılması, görsel araçlardan yararlanılarak öğrencinin motivasyonunun yüksek tutulması, birden fazla ortam araçlarının kullanılması bunlara örnek gösterilebilir. Programlamayı kolaylaştırma çalışmalarında, öğrenenlerin programlama mantığını daha iyi anlaması ve öğrendiklerini daha kalıcı olarak ilişkilendirebilmesine odaklanılır.  Bu düşünceden hareketle; programlamanın çalışma mantığı ile bireyin kendi eylemlerini ilişkilendireceği bir ortam tasarımı bireyin zihninde canlandırdığı durumu gerçek dünyada somut bir şekilde görmesine katkı sağlayabilir. Bu şekilde öğrenenler gerçek dünya ile etkileşime geçebilecekleri ve öğrenmelerinin kolaylaşabileceği değerlendirilebilir. Bu düşünceden hareketle bu çalışmada soyut kavramlarım somutlaştırmasına katkı sağlayan artırılmış gerçeklik (AG) uygulamalarının potansiyelinden yararlanılmıştır. Nitekim; AG uygulamalarının, anlamlı öğrenmeyi ve transferi kolaylaştırma noktasında eğitim sürecinde oldukça etkili olduğu, gözle görülemeyen soyut yapıları 3B olarak görselleştirerek içeriği somutlaştırdığı ve karmaşık konuları daha anlaşılır hale getirdiği belirtilmektedir. AG teknolojisi doğasındaki gerçek zamanlı etkileşimle anında dönüt sağlamakta ve öğrencileri öğrenme süreçlerini kontrol etmelerine imkân tanımaktadır. Bu potansiyeli değerlendirebilmek için içerik-teknoloji ilişkisi uygun biçimde ele alınmalıdır. Diğer bir ifadeyle programlama sürecindeki yapıların AG ortamına taşınması veya AG ile desteklenmesi gerekli durumların belirlenmesi önem arz etmektedir. Dolayısıyla içerik ve teknoloji hazırlanan senaryolar çerçevesinde bütünleştirilebilir. Bu çalışma, programlama sürecinde temel yapıları ve bu yapıların çalışma mantığını kavratmak için AG tabanlı bir öğrenme ortamı oluşturmada işe koşulacak öğretim senaryolarının özellikleri üzerine odaklanmıştır. Bu çalışmada programlamanın temel yapıları ve bu yapıların yürütülme mantığı temel alınarak geliştirilen senaryolar için programlama yapısı, senaryo bileşenleri ve AG teknolojisi ilişkisini ortaya koyan bir çerçeve önerilmiştir. Önerilen senaryoların geliştirilme ve uygulama aşamalarıyla programlama çalışmalarına özellikle programlama bilgisini bilgiyi transfer etme bağlamında katkı sağlayacağı düşünülmektedir.

İyi tasarlanmayan ve ders içinde nasıl kullanılacağı netleştirilmemiş AG araçlarının öğretim sürecinde kullanımı dezavantajları beraberinde getirebilir (Rasimah vd., 2011). Dolayısıyla AG teknolojisinin sınıf içi uygulamalara entegre edilmesi için AG teknolojisinin belli bir kuramsal çerçevede öğretimsel hedeflere uygun olarak ortamının tasarlanması ve derse entegre edilmesi önerilir (Rasimah vd., 2011). Bu bağlamda AG teknolojisine dayalı bir öğrenme ortamının tasarımının iyi olması AG teknolojisinin içerisinde bulunan 3B ve hareketli nesne özelliklerinin öğretimsel hedeflere uygun olmasına, bu hedefleri sunabilecek senaryoların üretilmesine ve gerektiğinde bilginin uygun biçimde arttırılarak anında öğrencinin öğrenme ihtiyacının karşılanmasına bağlıdır (Şahin, & Yılmaz, 2020). Dolayısıyla programlama öğretimine yönelik kullanılacak AG teknolojisinin başarısı, hazırlanacak AG ortamının şemalaştırmak için kullanıldığı öğretimsel sahnelere bağlı olacaktır.

**Yöntem**

Senaryo tasarımı, programlama temel yapılarının belirlenmesi, programlama sürecinin senaryoya dönüştürülmesi, her bir temel yapı için ayrı ayrı sahnelerin planlanması, 3B ortam geliştiricisi ve programlama alan uzmanı dönütlerinin alınması ve senaryoya nihai şeklinin verilmesi olmak üzere 5 aşamadan oluşmaktadır

*Programlama temel yapılarının belirlenmesi aşamasında* araştırmacı ve uzmanlar (U1, U2, U3) ile senaryo ile öğrencilere kazandırılması düşünülen kazanımlar ve bu kazanımlara denk gelen programlama temel yapıları eşleştirilmesi istenmiştir. *Programlama sürecinin senaryolaştırılması aşamasında* programlama eğitimi alan uzmanı (U3), 3B ortam geliştiricisi (U4), AG ortamında kullanılacak programlama yapıları, bu yapılara AG bileşenlerinin eklenebilirliği ve oluşacak senaryoların öğrenmeyi kolaylaştırması durumlarını değerlendirmeleri istenmiştir. *Programlama temel yapıları için ayrı ayrı sahnelerin planlanması aşamasında*, araştırmacı ve iki alan uzmanı (U1, U2) programlama temel yapılarına uygun sahneleri tasarlamıştır. *3B ortam geliştiricisi (U4) ve programlama alan uzmanı (U5) dönütlerinin alınması aşamasında* ise hazırlanan senaryolar programlama temel yapılarını ne derece yansıttığı ve programlamaya uygunluğu açısından programlama uzmanı ve bu senaryoların artırılmış gerçeklik teknolojisi üzerinden üzerinden geliştirilebilir olup olmadığı açısından 3B ortam geliştiricisi (U5, U4) tarafından değerlendirilmiştir. *Senaryoya nihai şeklinin verilmesi aşamasında* ise senaryolar sadeleştirilmiştir. Uzmanlardan elde edilen nitel veriler betimsel analizi yöntemi kullanılarak analiz edilmiştir.

**Bulgular**

AG'ye uygun senaryo geliştirilmesi sürecinde her bir aşama ayrı başlıklarda anlatılmıştır.

#### Programlama temel yapılarının belirlenmesi

Senaryolar tasarlanmadan önce, çalışma grubuna uygun programlama kazanımları MEB öğretim programından (MEB, 2018) “Bilişim Teknolojileri Dersi Programlama” konusu başlığı altında belirlenmiştir. Araştırmacı ve uzmanlar (U1, U2, U3) ile senaryo temelli AG ile öğrencilere kazandırılması düşünülen kazanımlar ve bu kazanımlara denk gelen programlama temel yapıları eşleştirilmiştir. Kazanımlar ve programlama konuları Tablo 6’da sunulmuştur.

Tablo . Kazanımlar ve Programlama Konuları

|  |  |
| --- | --- |
| Kazanımlar | Programlama Konuları |
| Programlamayla ilgili temel kavramları açıklar. | Değişken Tanımlama |
| Veri Operatörleri |
| Doğrusal mantık yapısını açıklar. | Ekrana yazdırma |
| Veri Okuma |
| Karar yapısını ve işlevlerini açıklar. | if/Koşul Yapısı |
| Karar yapıları içeren algoritmalar geliştirir. |
| Döngü yapısını ve işlevlerini açıklar. | Döngü Yapısı |
| Döngü yapısı içeren algoritmalar oluşturur. |
| Tüm programlama yapılarını içeren özgün bir proje oluşturur. | Değişken Tanımlama, Veri Operatörleri |
| Ekrana yazdırma, Veri Okuma |
| İf/Koşul Yapısı, Döngü Yapısı |

Tablo 6’da yer alan kazanımlara ilişkin programlama temel yapıları belirlenerek, yapılacak programlama öğretimi ve tasarlanacak Senaryoların tasarım sürecinin kapsamı oluşturulmuştur.

#### Programlama sürecinin senaryolaştırılması

Programlama sürecinin senaryolaştırılması, temelde her bir programlama yapısının ilgili kazanımlar çerçevesinde kullanılabileceği ve yazılan kodun AG’de gösterilebileceği bir hikâyeye dönüştürülmesini içerir. Bu süreç, alan uzmanları (U1, U2, U3) ve araştırmacının, senaryoda kullanılacak programlama yapıları, bu yapılara AG bileşenlerinin eklenebilirliği ve oluşacak senaryoların öğrenmeyi kolaylaştırması bağlamındaki değerlendirmeleri sonucunda şekillenmiştir. Bu değerlendirmeler sonucunda geliştirilecek olan AG temelli senaryoların özellikleri belirlenmiştir.

1. *Programlama akışı yapısının olması gereklidir.* Programlama süreci bir akıştan oluşmaktadır. Bu durumda programlama, başlangıç ve bitiş noktası olan ve bu iki nokta arasında birden fazla adımlar içeren ve birbirini etkileyen bir süreçtir. Dolayısıyla yazılan programlamanın senaryosu *bir akış* içermelidir.
2. *Programlama temel yapılarının öğrenmelerini kolaylaştırma amacıyla senaryoların programlamaya ilişkin şemaların kolaylıkla oluşturabilecek çerçevede tasarlanması gerekmektedir*. Alan yazında gerçek yaşam temeline dayanan bilgileri metaforlaştırılarak öğrenene sunmanın hem bireyin birden fazla duyusuna hitap edeceği hem de bu metaforlar sayesinde bilgiyi daha kolay organize etmesine yardımcı olacağı belirtilmektedir (Hurtienne (2017; Allbritton, 1995). Bu şekilde zihindeki önceki şemalar arasında ilişkiler geliştirilebilir ya da yeni şemalar oluşturulabilir.
3. *Programlama yaparken öğrenci kod ekleme ve çıkarma işlemi yapacağı için kullanılacak senaryo esnek olmalıdır*. Programlama birden fazla kod satırından oluşabilmektedir. Programlama problemine uygun olarak programlamanın içinde kod satırları eklenebilir, çıkarılabilir ya da program içerisindeki kodlar değiştirilebilir. Bu durum farklı kodların yazılabileceği esneklikte senaryolara ihtiyaç olduğuna işaret etmektedir.
4. *Senaryolar içerisindeki sahneler birbiriyle ilişkili olması gereklidir.* Programlama problemleri bir bağlam içerisinde çözüldüğünden çözüm sürecinin anlaşılırlığı için her bir yapının ayrı bağlamdan oluşması yerine birbiriyle ilişkili bir bağlam içinde olması gerekmektedir.

Senaryolar oluşturulurken dikkate alınan bu dört özelliğin “trafik teması” üzerinden metaforlaştırılmasının mümkün olabileceği programlama alan uzmanının (U5) değerlendirmeleri sonucunda belirlenmiştir. Akan bir yol *(süreç),* sürekli aktif ve dinamik *(esnek),* neredeyse her gün herkesin hayatında olan bir eylem *(günlük yaşam temelli),* ve birbirine bağlanan yollar *(ilişkili),* olduğundan Trafik temasının uygun olabileceği değerlendirilmiştir. Özellikle kavşaklar, trafik tabelaları, arabanın hareketliliğinin bir süreklilik sağlaması bu bağlamın programlama sürecini yansıtmasında etkili olacağı düşünülmüştür. Senaryo teması belirlendikten sonra uzmanlar (U1, U2) ve araştırmacı tarafından senaryo detaylandırılmıştır. Senaryoda tüm programlama süreci, bir noktadan diğer bir noktaya giden aracın trafik hareketleri ile, programlama sürecinde kodlanan her yapının işlevleri ise trafik nesneleri (araba, yol, sürücü, trafik lambası, tabela) ile ilişkilendirilerek görselleştirilmelidir. Kod yazılmaya başlandığında, artırılmış gerçeklik ara yüzünde bir araba ve kod içerisinde yazılan komutlara göre trafik nesneleri ve animasyonlar ilgili kod satırına göre oluşturulması örnek senaryo için tasarlanmıştır.

#### Programlama temel yapıları için sahnelerin hazırlanması

Senaryo öğrencilerin karşısına yazılan koda uygun sahneler şeklinde tasarlanmıştır. Bu sahneler belirlenirken programlama temel yapıları ayrı ayrı ele alınarak incelenmiş ve her bir yapıya uygun sahneler oluşturulmuştur. Dolayısıyla araştırmacı ve uzmanlar tarafından hazırlanan senaryo ilk olarak programlama temel yapılarını içeren 7 ana sahneye bölünmüştür. Örnek programlama temel yapıları ve bu yapılara uygun sahnelere ait senaryolar Tablo 7’de yer almaktadır.

Tablo . Programlama ekrana yazdırma, koşul yapıları ve bu yapılara uygun sahnelere ait senaryolar

|  |  |
| --- | --- |
| Programlama Temel Yapısı | Programlama yapılarının AG ortamında koda karşılık gelen sahne (artırılmış gerçeklik bilgisi) |
| Ekrana Yazdırma | * Kod penceresinde “write” komutu yazıldığında arabanın yanında kalem çıkacak “kalemde yazma animasyonu olacak” Write satırı bittiğinde araba duracak ve araba üzerinde bir mektup ikonu ortaya çıkacak ve bu ikon büyüyerek panele dönüşecektir. Ekrana yazılması istenen komut panelde çalışacaktır. Örneğin ‘merhaba dünya’ yazdırdı ise panel üzerinde “merhaba dünya” yazacaktır. Çarpı şeklinde kapat düğmesine tıklanacak ve ardından panel, mektup zarfına dönüşüp yukarıya gidip kaybolacak (herhangi bir veri alımı olmadığı için) ve araba yoluna devam edecektir. |
| * Ekrana yazdırma kodunun AG karşılığı ile her şey aynı olacak sadece veri girişi yapılacağı için veri girilmesi gerekecektir. Bu panele klavyeden bir şey yazılacak ve kare şeklindeki onay kutusuna tıklandıktan sonra panel mektup zarfına dönüşüp arabanın üstünde duracaktır. (Alınan veri değişkene aktarılacağı için) |
| Koşul/Şart Yapısı (İf-else) | * Öğrenci kod satırında if kod bloğunu yazdığında hareket halindeki araç trafik lambasıyla karışılacak ve duracaktır. * Trafik lambasını gördükten 4 5 saniye sonra yolu çubuk bir engel ile kaplı olan iki ayrı yol çıkacaktır. * Trafik lambasının elinde ünlem işareti şeklinde bir tabela olacak ve tabelanın yanında, üzerinde if kod bloğunun yazılı olduğu bir panel açılacaktır. Else yazdığında ise bu panelin sağ tarafına bu kod bloğu gelecektir. * Programlama esnasında if ve else komutu bittiğinde kıyaslama işlemi yapılacaktır. * Panelin içerisinde bulunan koda göre karakterin şapkalarında bulunan değerler kıyaslanacaktır. * Bu kıyaslama işlemi tıpkı işlem yapma modülündeki gibi şapkalar büyüyerek belli edilecektir. * İf komutunun içerisindeki değişkenlere ait karakterlerin şapkası arabadan yükselecek ve karşılaştırma işlemi yapılacaktır. * Karşılaştırmanın doğruluğu yeşil tik(if) işareti ile yanlışlığı(else) ise kırmızı çarpı işareti ile gösterilecektir. * Eğer tik işareti yanarsa trafik lambasında yeşil yanacak ve 1. Yoldaki engel kalkıp araba o yoldan devam edecektir. Eğer kırmızı çarpı işareti yanarsa trafik lambasında kırmızı ışık yanacak ikinci yoldaki engel kalkıp yola devam edilecektir. * 1. ve 2 yol olmak üzere iki ayrı yolun belli bir kısmı kendi yolunun rengine göre renklendirilecek olup uyarı levhasıyla karşılaştırma sonucu gösterilecektir. |

Tablo 7’de görüldüğü üzere her bir programlama temel yapısı 7 ayrı sahne olarak ayrıntılandırılmış ve her bir sahne trafik teması ile ilişkilendirilmiştir.

#### Senaryoların tasarıma dönüşmesi ile ilgili uzman değerlendirmeleri

Araştırmacı ve uzmanlar tarafından hazırlanan senaryoların programlama temel yapılarını ne derece yansıttığı ve programlamaya uygunluğu, programlama uzmanı tarafından senaryoların AG üzerinde geliştirilebilir olup olmadığı ise 3B ortam geliştiricisi tarafından değerlendirilmiştir.

Senaryolar üzerine programlama uzmanından alınan değerlendirmeler araştırmacı tarafından incelenmiştir ve 3 grupta toplanmıştır.

* Her bir temel yapıya ait oluşturulan sahnelerin birbiri içine geçmiş birden fazla sahneden oluşması
* Sahnelerin, programdaki kodun anlaşılması için fazladan bilişsel yük getirmesi
* Döngü yapısını anlatmak için tasarlanan sahnenin diğer yapılardan farklı bir bağlamda olması

Senaryoların AG ortamında geliştirilebilirliği üzerine 3B ortam geliştiricisi tarafından incelenmiş ve mevcut AG geliştirme araçlarının potansiyellerinin sınırlı olduğu bazı durumlar ortaya koymuştur. Bu durumlar:

* Değişken tür dönüşümüne ait sahnenin AG ortamına uyarlanması,
* Değişken tanımlama yapısına ait sahnenin AG ortamına uyarlanması,
* AG ortamının görüntülenmesi için kod satırı yazımının tamamlanması

şeklinde özetlenebilir.

Tüm bu durumlar, öğretimsel hedefler dikkate alınarak senaryo tasarımından çıkartılmıştır.

#### Senaryolara nihai şeklinin verilmesi

Uzmanlar ve araştırmacı tarafından gelen dönütler üzerine senaryolar sadeleştirilmiştir. Bu çerçevede programlama temel yapıları, sahne adları ve sahneler Tablo 8’da yer almaktadır.

Tablo . Senaryolara Ait Nihai Tablo

|  |  |  |
| --- | --- | --- |
| Programlama Temel Yapısı | Sahne Adı | Sahne |
| Değişken Tanımlama | Kutu | Hareket halindeki arabanın üzerinden değişken tanımlandığını ifade eden bir ok şeklinde tabelanın çıkması ve tanımlama bittikten sonra tanımlandığı değişkenin türüne göre rengi değişen ve üstünde verilen değişken adı yazan bir kutun olması araba hareket ettikçe o kutuda araba ile sabit bir şekilde gelmesi |
| Ekrana Yazdırma | Panel | Ekrana yazdırma yapıldığında ekrana bir şey yazıldığını gösteren arabanın üzerinde ok şeklinde bir tabelanın çıkması ve ekrana yazdırılan ifadenin araba üzerinden çıkan bir panele yazması |
| Veri Operatörleri | Panel | Veri operatörleri kullanıldığında atama operatörünün kullanıldığını gösteren arabanın üzerinde ok şeklinde bir tabelanın çıkması ve atama yapıldığını ve atama sonucunu araba üzerinden çıkan bir panele yazması |
| Veri Okuma | Buton | Veri okuma yapıldığında veri alınıyor ifadesini gösteren arabanın üzerinde ok şeklinde bir tabelanın çıkması ve okutulacak değeri girmek için bir değer giriniz şeklinde bir butonun çıkması |
| Koşul Yapısı(if) | Y kavşak | Koşul yapısı kullanıldığında koşulun yapıldığına dair bir tabelanın çıkması, ortaya bir ikiye ayrılan yolun çıkması, koşulun sağlanıp sağlanmadığını gösteren evet ve hayır tabelasının tam ikiye ayrılan yolun orda olması, uygun programlama koduna göre iki yoldan birinden gitmesi |
| Döngü (for) | Yuvarlak kavşak | Koşul yapısı kullanıldığında koşulun yapıldığına dair bir tabelanın çıkması, ortaya ikiye ayrılan yolun çıkması, koşulun sağlanıp sağlanmadığını gösteren evet ve hayır tabelasının tam ikiye ayrılan yolun orda olması, uygun programlama koduna göre iki yoldan birinden gitmesi koşul yapısının içinde gerçekleşen kıyaslama işleminin bir panelde yazılması |
| Döngü içinde Koşul | Yuvarlak içinde  Y kavşak | Döngü içinde koşul kullanıldığında ortaya yuvarlak yolun içinde ikiye ayrılan yolun çıkması ve koşula göre döngünün çalıştığını göstermesi her iki yapıdaki tüm özelliklerin bu bölümde de olması |
| Döngü İçinde Döngü | İçiçe  Yuvarlak kavşak | Döngü içinde koşul kullanıldığında ortaya yuvarlak yolun içinde bir yuvarlak yolun çıkması ve en içteki döngüye göre döngünün çalıştığını göstermesi her iki yapıdaki tüm özelliklerinde bu bölümde de olması |

**Sonuçlar**

Programlama sürecinde öğrenciler programlama becerisini sergilemek için birçok zihinsel etkinlik içerisine girerler. Bu süreçte bilginin bellekte işlenip yeni duruma aktarılması olarak değerlendirilen bilginin transferi, üst düzey düşünme becerileri arasında ele alınır. Üst düzey düşünme becerisi, öğrencilerin sadece hatırlamalarını değil aynı zamanda öğrendiklerini anlamalarını ve kullanabilmelerini gerektirir (Anderson, & Krathwohl, 2001). Bu kapsamda programlama öğretimini kolaylaştırmak için birçok öğretim yöntem ve teknikleri önerilmiş, gelişen teknolojiler çerçevesinde öğrencilerin kavramsal veya stratejik bilgiyi edinmeleri, bu bilgileri ilişkilendirmeleri amacıyla animasyon, simülasyon gibi görsel araçların yanı sıra sürükle bırak yardımıyla oyunlaştırılmış programlama uygulamaları, robotik kodlama, sanal gerçeklik ve artırılmış gerçeklik (AG) gibi teknolojilerden yararlanılmıştır (Esteves, & Mendes, 2004; Helminen, & Malmi, 2010; Kert, & Uğraş, 2009; Ozoran vd., 2012).

Alan yazındaki çalışmalar arasında programlama öğretiminde AG uygulamalarına yer veren çalışmaların sınırlılığı, AG’nin zihinsel süreçleri kolaylaştırma potansiyeli ve programlamanın doğası ilişkisinin kurulmasına ilişkin yeterli değerlendirmelerin olmayışına sebep olmaktadır. Bu doğrultuda bu araştırma AG’nin programlama sürecinde yaşana zihinsel süreçlerde gerçeğe yaklaştırma potansiyelini kullanarak öğrencinin programlamaya yönelik zihinsel model oluşturma süreci için harcayacağı çabayı azaltmayı hedeflemiştir. Dolayısıyla öğrenciler bu çabayı gerçeğe yakın model üzerinden mevcut şemaların devreye sokulması ve bellekte gerekli ilişkilerin kurulmasına harcayarak anlamlı öğrenme gerçekleştirmelerini sağlayacak senaryoların önemli olduğu görülmektedir (Balcisoy vd., 2000; Henderson & Feiner, 2009). Önerilen senaryoların geliştirilme ve uygulama aşamalarıyla programlama çalışmalarına özellikle programlama bilgisini bilgiyi transfer etme bağlamında katkı sağlayacağı düşünülmektedir.

**Kaynakça**

Anderson, L. W., & Krathwohl, D. R. (2001). *A taxonomy for learning, teaching, and assessing: A revision of BloomTs taxonomy of educational objectives*. New York: Longman's.

Esteves, M., & Mendes, A. J. (2004, October). A simulation tool to help learning of object oriented programming basics. In *34th Annual Frontiers in Education, 2004. FIE 2004.* (pp. F4C-7). IEEE.

Helminen, J., & Malmi, L. (2010, October). Jype-a program visualization and programming exercise tool for Python. In *Proceedings of the 5th international symposium on Software visualization* (pp. 153-162).

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

Ozoran, D., Cagiltay, N., & Topalli, D. (2012). Using scratch in introduction to programming course for engineering students. In *2nd International Engineering Education Conference (IEEC2012)* (Vol. 2, pp. 125-132).

Balcisoy, S., Kallmann, M., Fua, P., & Thalmann, D. (2000, October). A framework for rapid evaluation of prototypes with augmented reality. In *Proceedings of the ACM symposium on Virtual reality software and technology* (pp. 61-66).

Henderson, S. J., & Feiner, S. (2009, October). Evaluating the benefits of augmented reality for task localization in maintenance of an armored personnel carrier turret. In *2009 8th IEEE International Symposium on Mixed and Augmented Reality* (pp. 135-144). IEEE.