Eğitim ve Öğretim Araştırmaları Dergisi
Journal of Research in Education and Teaching
Kasım 2017 Cilt: 6 Sayı: 4 ISSN: 2146-9199

KODLAMA EĞİTİMİNDE GÖRSEL SÜRÜKLE-BIRAK PROGRAMLAMA DİLİ UYGULAMALARI İÇİN OTOMATİK BAŞARIM ANALİZ SİSTEMİ

Okt. Uğur Yıldız
Kocaeli Üniversitesi Enformatik Bölümü
uguryildiz@kocaeli.edu.tr

Prof. Dr. M. Melih İnal
Kocaeli Üniversitesi Enformatik Bölümü
minal@kocaeli.edu.tr

Özet
Kodlama, çeşitli sektörlerde çalışan personel ve uzmanlar için kritik düşünme, problem çözme, bilgi ve teknoloji okuryazarlığı gibi anahtar bir yetkinlik (beceri) olarak görülmektedir. Bilgisayar programlama ve kodlama becerilerinin artırılmasına yönelik kodlama eğitimleri ilköğretim seviyesinden üniversiteye kadar sayısı giderek artan birçok ülkenin eğitim müfredatlarında yer bulmaktadır. Bununla beraber birçok platformlarda görsel sürükle-bırak programlama dilleri geliştirildiği fakat ölçme-değerlendirme faaliyetleri için ilgili platformların zayıf kaldığı ve zorluklar barındırdığı gözlemlenmektedir.

Bu platformlarından birisi de Scratch yazılımından yeniden uyarlanarak JavaScript ile geliştirilmiş web tarayıcı temelli çalışan SNAP yazılımıdır. Bu çalışmada kodlama eğitiminin ölçme-değerlendirme faaliyetleri için eğitmenlerin problem tanımlayabildiği, problemin çözümünde kullanılacak görsel sürükle-bırak bileşenleri (blocks) sınırlayabildiği / şartlandırabildiği ve ilgili problemin çözüm çıktılarının belirlenen koşullara uygunluğunun ölçülebildiği SNAP temelli otomatik başarım analiz sistemi prototipi önerilmekte ve ihtiyaç duyulan muhtemel temel özellikleri tartışılmaktadır.

Anahtar Sözcükler: Görsel Sürükle- Bırak Programlama Dili, E-öğrenme, JavaScript.

AUTO PERFORMANCE ANALYSIS SYSTEM FOR VISUAL DRAG-DROP PROGRAMMING LANGUAGE APPLICATIONS IN CODING EDUCATION

Abstract
Coding is seen as an important competence for critical thinking, problem solving, information and technology literacy by staff and professionals working in various sectors. Coding trainings for increasing computer programming and coding skills are found in educational curricula of many countries which are increasing in number from elementary school to university. However, it has been observed that visual drag-and-drop programming languages have been developed on many platforms, but the corresponding platforms for measurement and evaluation activities are weak and have difficulties.

One of these platforms is SNAP software which is based on web browser developed by JavaScript and readapted from Scratch software. In this study, it is proposed a SNAP based automatic performance analysis system prototype which can identify problems for instructor's problem and can be limiting / conditioning visual drag-and-drop components (blocks) to be used for solution of the problem and can measure the conformity of the solutions of the problem to the determined conditions, and discusses possible basic features needed.

Keywords: Visual Drag-Drop Programming Languages, E-Learning, JavaScript.

GİRİŞ

Kodlama eğitimlerinde, problemlerin çözümünün ilk adımı gerekli işlemlerin gerçekleştirilme sıralamasının (akışının) mantıksal bir düzen içerisinde tasarlanması yani algoritmasının

oluşturulmasıdır. Algoritma tasarımına dair uygulama ve etkinlikler günümüzde yaygın olarak görsel sürükle-bırak (GSB) programlama dilleri ile gerçekleştirilmektedir. Bu diller çeşitli kategorilerde (kontrol, döngü, mantık vb.) işlemlerin blok (block) adı verilen temsili görsellerle ifade edilmesi ve ilişkilendirilmesi temeline dayanmaktadır. Bir problemin çözümünde ihtiyaç duyulan araçları (işlemleri) temsil eden görsel nesneler birbirlerinin ardı sıra veya iç içe belirli bir düzende sıralanarak algoritmanın gerçeklenmesi sağlanır.

Tasarlanan algoritmanın istenilen işi yaptığına dair ölçme ve değerlendirmenin otomatik yapılması kodlama eğitimlerinde özellikle uzaktan eğitim platformlarında önemli bir ihtiyaç olarak ortaya çıkmaktadır. Bununla beraber, temelde kodlama eğitiminde belirli problemlerin istenilen araçlarla (işlemlerle) çözülmesi de beklenir. Kodlama eğitimlerinde GSB programlama dillerinin editörleri tüm araçları sunduğu için bu aşamada ilgili problemin çözümünde öğrenciler farklı araçlar kullanabilir. Bu durum istenilen becerinin kazandırıldığına dair gerçekleştirirlen ölçme ve değerlendirme faaliyetlerini olumsuz etkilemektedir.

GSB programlama dilleri genellikle web ortamında geliştirilmekte ve çoğu açık kaynak kodlu olarak yayınlanmaktadır. Bu çalışmada prototiplenecek otomatik başarım analiz sistemi de bütünleşik çalışma gerekliliği nedeniyle web tabanlı olarak tasarlanmıştır. Sistemin yönetim arayüzü geliştirme süreçlerini hızlandırmak için Yii2 geliştirme çatısı (framework) kullanılarak gerçeklenebilir (Yii PHP Framework, Web 2.0'da Uygulama geliştirmede en iyisi). Modüler yapıda JavaScript ile geliştirilmiş SNAP yazılımı sistemin görsel-sürükle bırak programlama dili editörü olarak tercih edilmiştir.

GÖRSEL SÜRÜKLE-BIRAK PROGRAMLAMA DİLLERİ

Günümüzde birçok GSB programlama dili geliştirilmiştir. Bu kapsamda ilk örnekler masaüstü uygulama olarak geliştirilmiş, fakat web teknolojilerinin yaygınlaşması ve geliştirme imkanlarının artması nedeniyle platform ve diller web temelli olarak tercih edilmeye başlanmıştır. Dünya genelinde yaygın olarak kullanılan Scratch platformu 2018'in ilk çeyreğinde HTML5 temelli 3. nesil sürümünü yayınlayacağını duyurmuştur (Scratch 3.0, Scratch Wiki). Bu projelerin genellikle açık kaynak lisansıyla dağıtılması, projelerin çatallanarak farklı uygulama alanlarına dair özelleşmelerinin de önünü açmıştır.
Kullanıcı arayüzlerinin ortak özelliği işlevlere karşılık gelen görsel simgelerden oluşan araç kutuları barındırmalarıdır. Bu araç kutuları işlevleri Hareket, Görünüm, Ses, Veri, Olay, Kontrol, İşleç ve Algılama şeklinde sınıflandırılmış olarak sunarlar. Bununla beraber çoğu editör özelleştirilmiş blok (block) oluşturma imkanı da sunmaktadır. Bu sayede çeşitli kod prosedürleri yazılabildiği gibi robot benzeri harici donanımların özel işlevleri de temsil edilebilmektedir.

GSB programlama dili editörleri eğitim amaçlı olarak çeşitli uygulama konseptlerini kullanmaktadırlar. Bunların başlıcaları aşağıda sıralanmıştır:

- Sahne Üzerinde Animasyon / Oyun Tasarımı
- Kod Üretimi / Uygulama Geliştirme
- Harici Donanım Kontrolü

Sahne Üzerinde Animasyon / Oyun Tasarımı

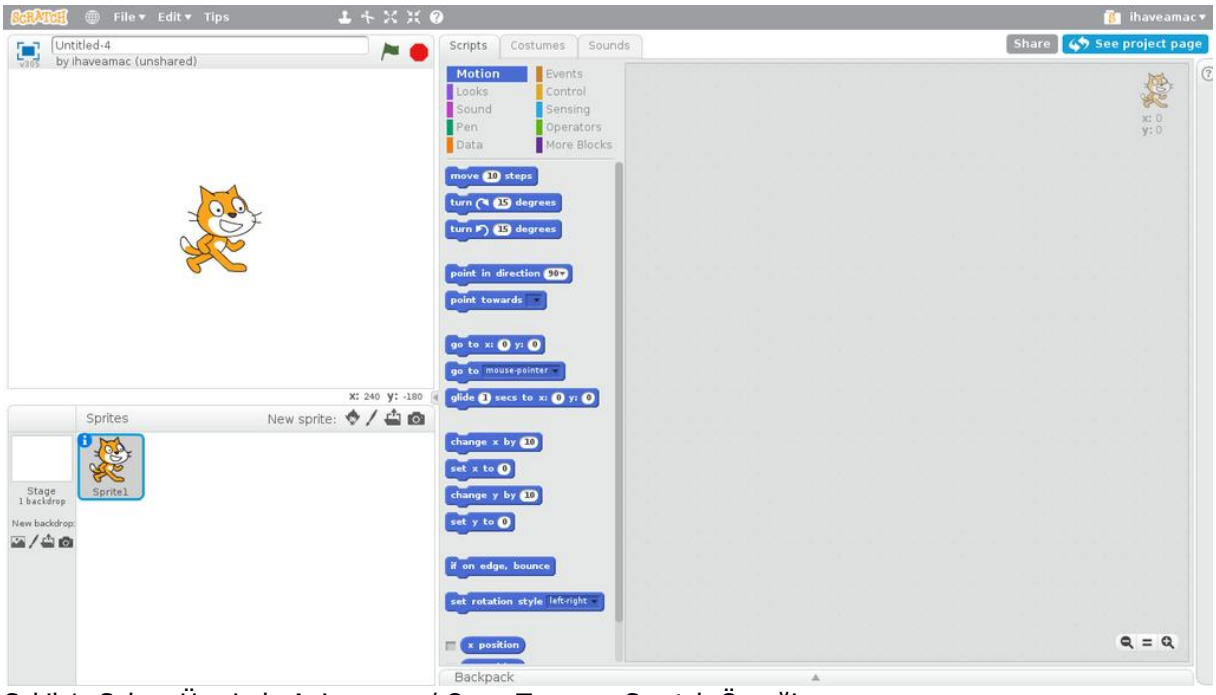
Şekil 1'de örneği görülen bu tip editörler bir araç kutusu, bir kukla yönetim paneli, bir sahne ve algoritma tasarımının gerçekleştiği bir çalışma alanından oluşmaktadır. Öğrenci araç kutusundaki işlevleri mantıksal bir düzen içerisinde çalışma alanına yerleştirerek, kuklaları ve sahnenin genelini kontrol edebilmektedir.

Kod Üretimi / Uygulama Geliştirme

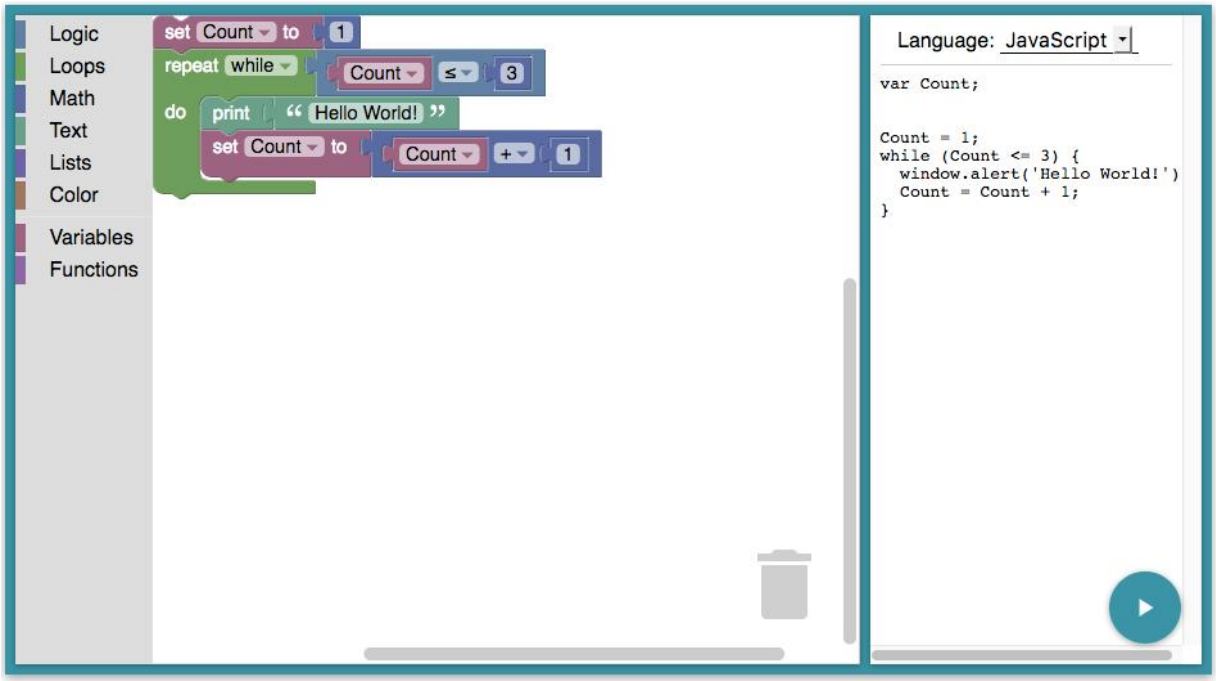
Kod üretimine yönelik editörlerde genellikle bir araç kutusu, tasarımın gerçekleştirildiği bir çalışma alanı ve üretilen kod çıktısının gösterildiği kod üretici paneli bulunmaktadır. Şekil 2'de örneği görülen bu tip uygulamalar kod ürettiği gibi üretilen kodların derlenerek çeşitli platformlarda koşturulabilmesi için çalışan/kurulan dosyalar da (örn. Android için apk) üretebilmektedir. Bu konseptteki uygulamaların çoğu Google tarafından geliştirilen Blockly kütüphanesi ile geliştirilmiştir. Blockly PHP, Javascript, Lua, Dart ve Python dillerinde kod üretebilmektedir (Blockly, İşlevleri).

Harici Donanım Kontrolü

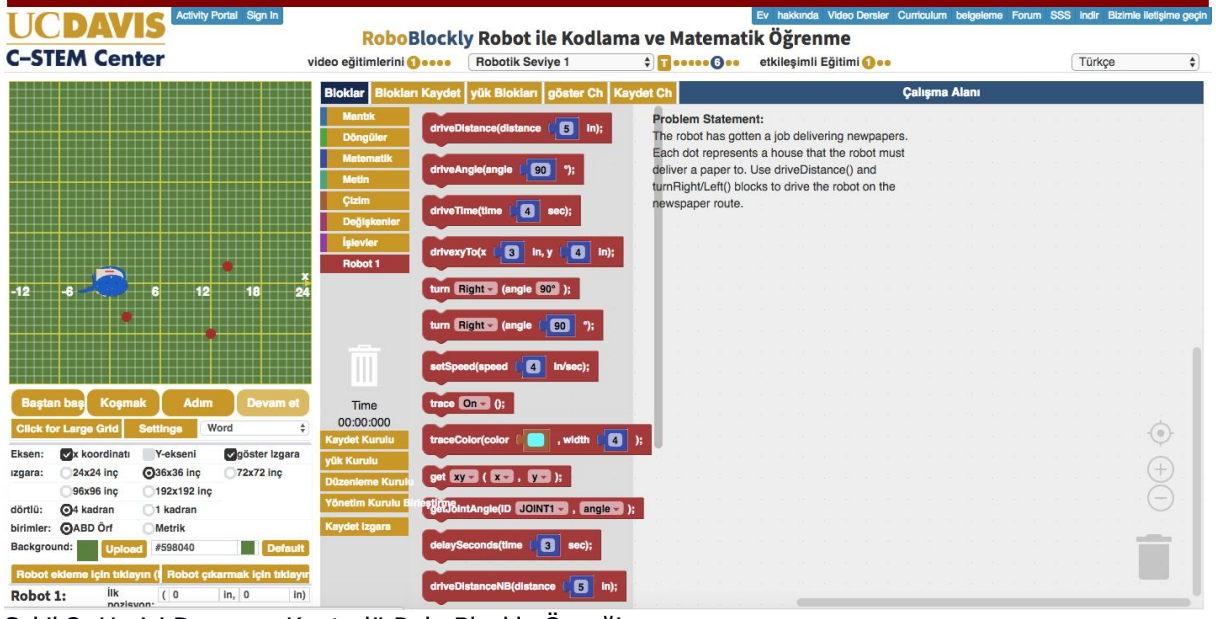
Blockly gibi kütüphaneler veya Scratch gibi GSB programlama dilleri temel alınarak çeşitli harici cihazların (Lego, MicroBit, Arduino vb) üzerinde koşacak algoritmalar tasarlanabilen editörlerdir. Şekil 3'te örneği görülen bu tip editörler temel kategorilerdeki işlevlerin yanı sıra geliştirildikleri harici donanıma özgü özel işlevler de barındırmaktadırlar. Genellikle cihazın simülasyonu için bir sahne, bir araç kutusu ve tasarım için çalışma alanı barındırmaktadırlar.



Şekil 1: Sahne Üzerinde Animasyon / Oyun Tasarımı Scratch Örneği



Şekil 2: Kod Üretimi / Uygulama Geliştirme Blockly Örneği

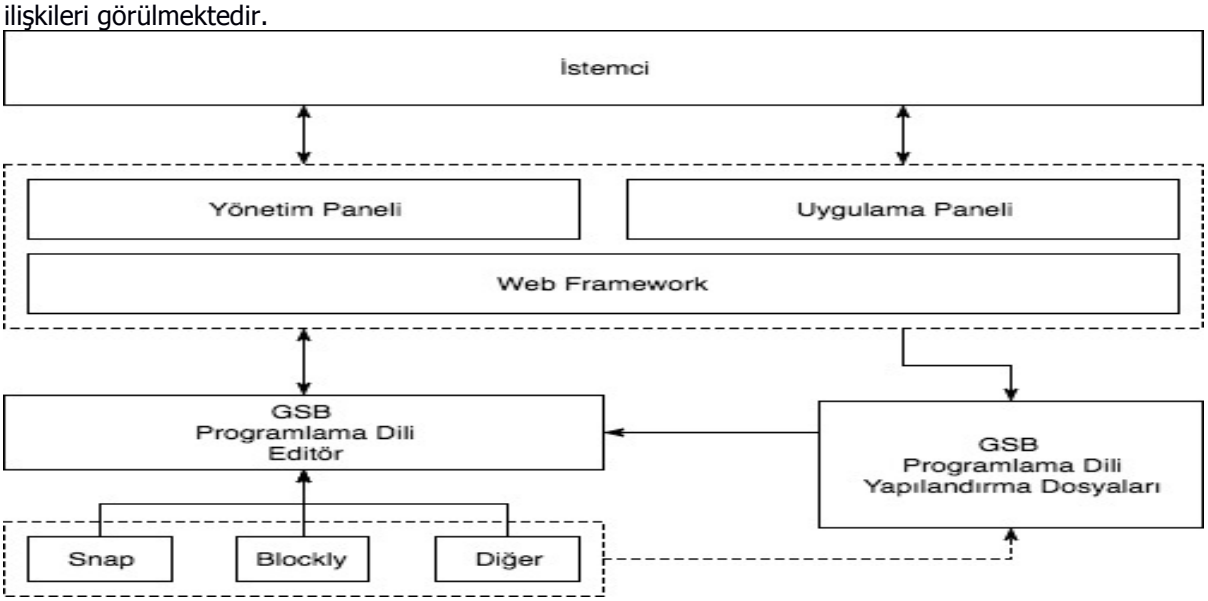


Şekil 3: Harici Donanım Kontrolü RoboBlockly Örneği

Mimari Tasarım

Genellikle MVC (Model-View-Controller) mimari ile geliştirilmiş olan GSB programlama dilleri, kullanıcı arayüzünün yapılandırılmasını ve çeşitli çalışma zamanı ayarlarını xml veya json temelli veri dosyalarını da barındırmaktadırlar. Bu yapı yeni uygulamalar türetilmesi/geliştirilmesi için esneklik sunmaktadır. Bununla beraber GSB programlama dilleri genellikle API (Application Programming Interface/ Uygulama Programlama Arayüzü) sunmaktadır. Geliştirilecek veya türetilecek yeni uygulamalar bu API'lar sayesinde kod çatallanmalarına rağmen çekirdek projenin güncellemelerini kendi çözümlerine dahil edebilmektedirler.

Geliştirilecek olan sistemin etkinlik/uygulama özelinde yapılandırılması için ilgili xml ve json dosyalarının dinamik olarak üretilmesi öngörülmektedir. Bununla beraber sunulan API'lar sayesinde çalışma esnasında kullanılan bileşenler ve nesnelerin izlenmesi de mümkün olabilmektedir. Bu izlemelerden, ölçme ve değerlendirme için gerekli tüm verilerin de elde edilebileceği değerlendirilmektedir. Şekil 4'te tasarlanan mimari yapıdaki bileşenler ve bileşenlerin birbirleriyle olan ilişkileri görülmektedir.



Şekil 4: Mimari Tasarım

PROGRAMLAMA EĞİTİMİNDE ÖLÇME VE DEĞERLENDİRME

Klasik programlama eğitiminde genellikle algoritmanın gerçeklenmesi eğitiminin verildiği dilde yazılmış olan program parçacıklarının debug edilmesi veya kodun gözden geçirilmesi ile değerlendirilmektedir. Aynı yöntem GSB programlama dilleri ile geliştirilen uygulamalarda da yaygın olarak kullanılmaktadır. Ayrıca GSB programlama dilleri ile etkileşimli eğitim içerikleri de hazırlanabilmekte fakat bu tip içeriklerin geliştirilmesi uzmanlık gerektirmekle beraber sınırlı örnek ve uygulama açısından da zayıf ölçme ve değerlendirme imkanı sunmaktadır.

Özellikle teknik açıdan uzman olmayan eğitimcilerin, GSB programlama dilleri ile yapılacak kodlama eğitimlerinde laboratuvar uygulamaları için esnek ve özelleştirilebilir içeriklere ihtiyaç duydukları görülmektedir. Aynı zamanda birebir eğitimci kontrolünde olmayan özellikle uzaktan eğitimle gerçekleştirilen çalışmalarda eğitimciden bağımsız asenkron ölçme ve değerlendirme ihtiyacı da gözlemlenmektedir.

Kodlama eğitimlerinde ölçme-değerlendirme faaliyetleri için eğitmenlerin problem tanımlayabilmesi, problemin çözümünde kullanılacak GSB bileşenlerini (blocks) sınırlayabilmesi/şartlandırabilmesi ve ilgili problemin çözüm çıktılarının belirlenen koşullara uygunluğunun ölçülebilmesi ihtiyaç duyulan özellikler olarak öne çıkmaktadırlar.

OTOMATİK BAŞARIM ANALİZ SİSTEMİ

Ölçme ve değerlendirme için geliştirilecek otomatik başarım analiz sisteminin yukarıda bahsedilen özellikleri barındırması ve geleceğe dönük olarak farklı GSB programlama dilleri için ortak bir altyapı sunması hedeflenmektedir. Sistemlerin entegrasyonu ve genel çalışma prensibi için önerilen mimari yapı aşağıda detaylı olarak incelenmiştir.

Sistemde tüm yönetim arayüzleri web geliştirme çatısının sunduğu altyapı ile modüler olarak gerçeklenebilir. Temelde yetkileri farklı olan en az iki rol tanımlanabilmelidir. Eğitimciler ve öğrenciler iki farklı senaryo ile sistemi kullanabilmelidirler. Eğitmenlerin temel kullanım senaryosu şu şekilde öngörülmektedir;

1. Yönetim panelinde problem tanımlar.
2. Çözüm için kullanılacak GSB programlama dilini belirler.
3. Problemin çözümü için kullanılacak araçların etkin/pasif olma durumlarını belirler.
4. Genel veya araç bazlı olarak işlevlerin kullanım adetlerini sınırlar.
5. Çözüm koşturma sınırını belirler.
6. İzlenecek değişken ve çıktılara dair değerlendirme/ölçme için gerekli koşulları belirler.
7. Problemi sistemde öğrencilerin erişimine açar.

Burada tanımlanan probleme ait tüm veriler veritabanı sisteminde tutulabilmelidir. Probleme özgü yapılandırma dosyaları çalışma zamanında üretilerek öğrencilerin kullanımına sunulabilir. Öğrencilerin kullanım senaryosu da şu şekilde öngörülmektedir;

1. Çözülecek probleme giriş yapar. (gerekli editör ve yapılandırma dosyaları otomatik oluşturulur)
2. Editör üzerinde tanımlanan problemin çözümü için gerekli işlevleri çalışma alanına sürükleyip bırakarak algoritma tasarımını gerçekler.
3. Tasarım bitiminde uygulamayı koşturur (çalıştırır), çözüme erişilme durumuna göre 2. adıma geri dönülür veya devam edilir.
4. Çözümü kaydeder ve gönderir. (sistem otomatik olarak izlediği değişken ve çıktılara göre sonuçları veritabanına kaydeder.
5. Bir problem farklı zamanlarda birden çok kez çözülebilir. Tüm çözüm girişimleri kayıt altına alınabilmelidir.

Bu bilgiler ışığında prototiplenecek olan sisteme dair şu özellikler önerilebilir;

- **Problem Tanımlama:** Eğitimci problemi tanımlayabilmek için açıklayıcı metin ve görseller girebilmelidir.
- **Sınırlanabilir Araç Kutusu:** Eğitimci problemin çözümü için gerekli araçları seçebilmeli ve alternatif çözümlere imkan sağlayacak araçları yasaklayabilmelidir.
- **Araç Kullanım Adeti Sınırlaması:** Problemin çözümünde öğrencinin istenilen beceriyi kazanabilmesi ve çözüme yönlendirilmesi için eğitimci kullanılacak araçların kullanım sayılarını sınırlandırabilmelidir.
- **Çözüm Koşturma Sayısı:** Problemin çözümü için oluşturulan algoritmanın doğru çözüme ulaşılana dek kaç kez koşturulduğu bilgisi tutulabilmelidir.
- **Değişken ve Çıktı İzleme:** Değişken ve çıktılar belirli koşullara bağlı olarak ölçülebilmeli/değerlendirilebilmelidir.

SONUÇ ÖNERİLER

Görsel sürükle-bırak programlama dillerinin uygulama alanlarının gittikçe yaygınlaştığı günümüzde bu araçlarla yapılan akademik amaçlı eğitim çalışmalarına dair başarımların otomatik olarak ölçülmesi de önemli bir araştırma konusu olarak öne çıkmaktadır. Bununla beraber eğitim içeriklerinin zenginleştirilmesi için; eğitimcilerin platforma dair geliştirme dillerinde uzman olmadan çeşitli laboratuvar uygulamaları geliştirebildikleri oluşturucu (builder) tipi sistemlere ihtiyacın da ortaya çıktığı gözlemlenmektedir.

Bu çalışmada günümüz web teknolojileri ile bu tip yönetim sistemlerinin ve GSB programlama dilleri entegrasyonunun gerçekleştirilebilmesine engel herhangi bir bulguya rastlanmamıştır. Hatta platformların açık kaynak kodlu olarak seçilmesi durumunda bu entegrasyonun daha da kolay olacağı öngörülmektedir.

Başarım analiz sisteminde oluşturulan uygulamaların taşınabilir olması, raporlama uyumluluklarının SCORM vb. standartları gözetmesi ve özellikle bu tip başarım analiz sistemlerinin yaygın olarak tercih edilen eğitim içerik yönetim sistemlerine entegre modüller olarak tasarlanabilmesi gibi başlıklar gelecek dönem çalışmalar olarak öne çıkmaktadırlar.

Not: Bu çalışma 26-27 Ekim 2017 tarihlerinde Antalya'da düzenlenen 6'ncı Eğitim ve Öğretim Çalışmaları Dünya Kongresi'nde bildiri olarak da değerlendirilmiştir.

KAYNAKÇA

Yii PHP Framework, Web 2.0'da Uygulama geliştirmede en iyisi, 06.11.2017 tarihinde http://www.yiiframework.com/ adresinden alınmıştır.

Scratch 3.0, Scratch Wiki, 06.11.2017 tarihinde https://wiki.scratch.mit.edu/wiki/Scratch_3.0 adresinden alınmıştır.

Blockly, İşlevleri, 06.11.2017 tarihinde https://developers.google.com/blockly/ adresinden alınmıştır.