
Orijinal Makale Başlığı:

Algoritma ve akış şeması kavramlarının öğretiminde akıllı bir yazılım sistemi kullanımı

Makalenin İngilizce Başlığı:

Usage of an intelligent software system in teaching algorithm and flowchart concepts

Yazar(lar):

Utku KÖSE, Aslıhan TÜFEKÇİ

Kaynak Gösterimi İçin:

Köse, U., & Tüfekçi, A. (2015). Algoritma ve akış şeması kavramlarının öğretiminde akıllı bir yazılım sistemi kullanımı. *Pegem Eğitim ve Öğretim Dergisi*, 5(5), 569-586, <http://dx.doi.org/10.14527/pegegog.2015.031>.

Original Title of Article:

Algoritma ve akış şeması kavramlarının öğretiminde akıllı bir yazılım sistemi kullanımı

English Title of Article:

Usage of an intelligent software system in teaching algorithm and flowchart concepts

Author(s):

Utku KÖSE, Aslıhan TÜFEKÇİ

For Cite in:

Köse, U., & Tüfekçi, A. (2015). Algoritma ve akış şeması kavramlarının öğretiminde akıllı bir yazılım sistemi kullanımı. *Pegem Eğitim ve Öğretim Dergisi*, 5(5), 569-586, <http://dx.doi.org/10.14527/pegegog.2015.031>.

Algoritma ve Akış Şeması Kavramlarının Öğretiminde Akıllı Bir Yazılım Sistemi Kullanımı

Utku KÖSE^{*a}, Aslıhan TÜFEKÇİ^b

^aUşak Üniversitesi, Bilgisayar Bilimleri Uyg. ve Arş. Mrk., Uşak/Türkiye

^bGazi Üniversitesi, Gazi Eğitim Fakültesi, Ankara/Türkiye



Makale Bilgisi

DOI: 10.14527/pegegog.2015.031

Makale Geçmişi:

Geliş 16 Temmuz 2015
Düzeltilme 03 Eylül 2015
Kabul 18 Kasım 2015

Anahtar Kelimeler:

Bilgisayar programlama,
Algoritma,
Akış şeması,
Yapay zekâ,
Eğitsel yazılım sistemi.

Öz

Bu çalışmanın amacı, bilgisayar programlamanın önemli konuları: algoritma ve akış şeması kavramlarının öğretilmesi için geliştirilmiş olan, Yapay Zekâ destekli bir eğitsel yazılım sistemini tanıtmak ve sistemin başarımını değerlendirmek için elde edilen bulgulara değinmektir. Çalışma kapsamında tanıtılan yazılım sistemi, bilgisayar programlama temel kavramlarının öğretimi düsturuna sıkı bir şekilde bağlı kalmakta, ancak bunu Yapay Zekâ destekli, akıllı bir mekanizma çerçevesinde gerçekleştirmektedir. Yazılımın etkili bir öğretim aracı olup olmadığı konusunda fikir edinmek için genel bir değerlendirme süreci planlanmış; çalışmaya konu olan öğrenciler, bu süreçten geçirilmiştir. Değerlendirme süreciyle elde edilen bulgular, geliştirilen yazılım sisteminin, algoritma - akış şeması kavramlarını ve bilgisayar programlamanın temel odak noktası olan algoritmik düşünce mantığını etkili bir şekilde öğretilmesi noktasında başarılı olduğunu göstermiştir. Ek olarak; yazılım sisteminin, öğrencilerin bilgisayar programlama temellerine yönelik derslerdeki başarı oranlarını artırdığını ve gerek yazılımın, gerekse gerçekleşen eğitimsel süreçlerin, öğrenciler tarafından oldukça etkili bulunduğunu da ifade etmek mümkündür.

Usage of an Intelligent Software System in Teaching Algorithm and Flowchart Concepts

Article Info

DOI: 10.14527/pegegog.2015.031

Article history:

Received 16 July 2015
Revised 03 September 2015
Accepted 18 November 2015

Keywords:

Computer programming,
Algorithm,
Flow chart,
Artificial intelligence,
Educational software system.

Abstract

Objective of this work is to introduce an Artificial Intelligence supported educational software system, which has been developed for teaching important subjects of computer programming: algorithm and flowchart concepts, and touch upon the findings, which were obtained for evaluating success of the system. The software system introduced in the work is tightly connected to the rule of teaching essential computer programming concepts, but ensures this task in the context of an Artificial Intelligence supported, intelligent mechanism. In order to have idea about whether the software is an effective teaching tool or not, a general evaluation process has been planned; students subjected to the work have been taken into this process. Findings obtained via evaluation process have shown that the developed software system is successful at effectively teaching the algorithmic thinking logic, which is the essential focus, and algorithm - flowchart concepts. Additionally, it is also possible to express that the software system has improved students' success rates in the courses related to essentials of computer programming and students have found both software and the performed educational processes pretty effective.

*Yazar: utku.kose@usak.edu.tr

Giriş

Bilgisayar programlama, bilgisayar tabanlı sistemlerin istekler doğrultusunda çalışması ve problemleri çözme konusunda yönlendirilmesi için sahip olunması gereken temel bir beceridir. Günümüz teknolojisinde meydana gelen hızlı gelişme ve iyileşmeler, bilgiye olan ihtiyaçların artması ve ihtiyaç duyulan bilgiye hızlı erişim sağlayabilme gibi değişimler, bilgisayar programlamaya yönelik temel düzeyde bilgi ve becerilerin farklı araştırma / bilim / faaliyet alanlarında da gerekli bir hale gelmesine sebep olmuştur. Hatta son zamanlardaki gelişmeler, bilgisayar programlama yaklaşımının okul ortamlarına yayılma hızının arttığını göstermektedir (Kafai & Burke, 2015; Sterritt et al., 2015). Bilgisayar programlamaya yönelik artan bu ilgi, ilgili bilgi ve becerilerin kazandırılması yönünde eğitimsel süreçlerin de birçok çalışmaya konu olmasını sağlamıştır. Gerçekleştirilen çalışmalar, bilgisayar programlamanın öğrenilmesi ve öğretilmesi noktasında dönüm noktası teşkil eden tecrübeleri ön plana çıkarmıştır. Bilgisayar programlama öğretiminde söz konusu olan en önemli tecrübe, hiç kuşkusuz ki algoritma ve akış şeması kavramlarının öğrenilmesi süreci olmaktadır. Bu nedenle özellikle bu kavramların etkin öğretilmesi üzerine kurgulanan çalışmalar, son yıllarda oldukça rağbet görmektedir.

Konuya ilişkin literatür incelendiğinde, bilgisayar programlamanın öğretilmesi / öğrenilmesi noktasında çözülmesi gereken farklı sorunlar olduğu da görülmektedir. Özellikle problem çözme yaklaşımları ve çözüm tasarlayabilme yeteneklerinin etkin düzeyde elde edilememesi, bilgisayar programlama becerisinin öğrenilmesi zor bir yaklaşım olarak kabul edilmesine sebep olabilmektedir (Moser, 1997; Pillay, 2003; Pillay & Jugoo, 2005). Bu sorunun üstesinden gelmek adına programlama becerisinin anlaşılması, farklı yollarla kazandırılması ve bilgisayar programlamaya temel teşkil eden derslerdeki katılım oranı düşüşlerinin ortadan kaldırılabilmesi yönünde çalışmalar gerçekleştirilmiştir (Bau et al., 2015; Bennedsen & Caspersen, 2007; Cañas et al., 1994; Cutts et al., 2014; Esteves et al., 2011; Krpan et al., 2015; Michaelson, 2015; Robins et al., 2003; Wiedenbeck et al., 2004; Wiedenbeck et al., 1999; Yadin, 2011). Bu bağlamda, gerçekleştirilen çalışmalar neticesinde ortaya çıkan önemli bir eğilim de, kullanımı kolay, görsel açıdan zengin ve etkileşimli öğretim ortamlarının literatüre kazandırılması olmuştur. Örneğin, Massachusetts Teknolojisi Enstitüsü'nde geliştirilen Scratch programı, bilgisayar programlamanın temelleri niteliğindeki algoritmaların ve algoritmik düşünce mantığının öğretilmesi adına etkin bir araç olarak kullanılmaktadır (Armoni et al., 2015; Maloney et al., 2010; Resnick et al., 2009). Yine ilgili literatürde, görsel süreçlerden yararlanan ve bilgisayar programlama ve algoritmik düşünce mantığı öğretilmesine odaklanan farklı uygulamalar da söz konusudur (Carlisle et al., 2005; Chen & Morris, 2005; Hooshyar et al., 2015). Konuya ilişkin olarak değinilmesi gereken diğer bir husus da, Yapay Zekâ yaklaşım, yöntem ve tekniklerinin eğitimsel yazılım sistemleri tasarlanıp geliştirilmesinde aktif olarak kullanılıyor olmasıdır. Dolayısıyla, görsel ve etkileşimli kullanım özelliklerinin ötesinde, akıllı organizasyon, değerlendirme ve kullanım özelliklerinin işe koşulabilmesine imkân sağlayan Yapay Zekâ desteği, eğitimcilerin ilgi alanı içerisinde olmaya devam etmektedir (Dorça, 2015; Jain et al., 2013; Matsuda et al., 2015; Wenger, 2014).

Bilindiği üzere, bilgisayar programlama mantığının öğretimi aşamasında yüksek etkileşimli ve pratik uygulama süreçleri sunmayı amaç edinen farklı yazılım sistemleri araştırmacılar tarafından, özellikle son yıllarda sıklıkla geliştirilmektedir. Genel olarak incelendiğinde, görsel işlevselliği yüksek nitelikte olan ve kullanıcıların kod parçalarından ziyade; görsel kullanım mantığına dayalı elementleri kullanarak uygulamalar oluşturabildiği ve nihayetinde algoritmik akışı daha kolay algılayabildiği bu yazılımlar, eğitim kurumları tarafından büyük rağbet görmektedir.

Gerçekleştirilen bu çalışmada ortaya konulan yazılım sistemi, bilgisayar programlama temel kavramlarının öğretimi düsturuna sıkı bir şekilde bağlı kalmakta, ancak bunu Yapay Zekâ destekli, akıllı bir mekanizma çerçevesinde gerçekleştirmektedir. Bu bağlamda çalışmanın amacı; "bilgisayar programlamanın önemli konuları: algoritma ve akış şeması kavramlarının öğretilmesi için geliştirilmiş olan, Yapay Zekâ destekli bir eğitsel yazılım sistemini tanıtmak ve sistemin başarımını değerlendirmek için elde edilen bulgulara değinmek" şeklindedir. Alt-amaçları şöyle ifade etmek mümkündür:

- Akıllı yazılım sistemini kullanmış olan deney grubundaki öğrenciler ile mevcut, geleneksel eğitim süreci kapsamında yer almış olan kontrol grubundaki öğrencilerin akademik başarıları arasında farklılık var mıdır?
- Akıllı yazılım sistemini kullanmış olan deney grubundaki öğrencilere sistem ortamındaki problemlerden uygun olanlar, akıllı bir biçimde sunulabilmekte midir?
- Akıllı yazılım sistemini kullanmış olan deney grubundaki öğrenciler, derslerinde böyle bir yazılım sisteminin kullanılması konusunda olumlu fikirlere sahip midir?

Algoritma ve Akış Şeması Öğretiminde Akıllı Yazılım Sistemi

Çalışmada ortaya konulan; akıllı yazılım sisteminin eğitimsel anlamda amacını ve katkılarını anlamak adına; geliştirilen Yapay Zekâ mekanizmasını ve yazılım sisteminin kullanım yapısını ele almakta fayda vardır. Bu bağlamda, yazılım sisteminin Yapay Zekâ mekanizmasına ve kullanım özellik - işlevleri bu bölüm altında kısaca açıklanmaktadır.

Yapay Zekâ Mekanizması

Yazılım sisteminde öğretmenler sisteme algoritma ve/veya akış şeması problemlerini tanımlamakta ve muhtemel çözümleri tasarlamaktadır. Bu noktada her bir problem için 1 ile 5 arasında zorluk düzeyleri belirlenmektedir. Muhtemel çözümlere göre öğrencilerin tasarladıkları algoritma / akış şemalarında şu kıstaslara göre puanlama yapılmaktadır:

- Uygun şart - koşul yapılarının kurulması,
- Uygun döngü yapılarının kurulması,
- Algoritma ve akış şeması dallanmalarının doğru yapılması,
- Algoritma ve akış şeması tanımlamalarının doğru yapılması.

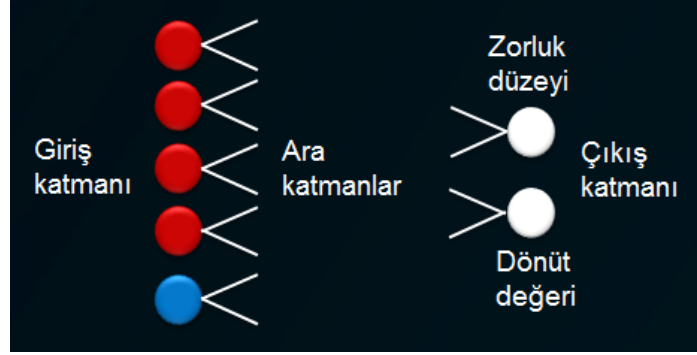
Belirlenen her kriterin farklı puan aralıklarını hedef alan çeşitli dönütler de (mesaj, örnek resim, video, Web sayfası, doküman gösterimi... vb.) yine öğretici tarafından sisteme tanımlanmaktadır. Öğreticiler yine sistem üzerinde tanımladıkları ufak test sınavları ile öğrenci başarıyı ölçebilmektedir.

Öğrencilerin gerçekleştirmiş olduğu problem, test çözümü... vb. aktiviteler sonrası farklı kategorilerde elde ettikleri puanlar bir Yapay Sinir Ağları sistemine giriş olarak verilmektedir. Buna göre, girişler kısaca şu şekildedir:

1. Giriş: Uygun şart - koşul yapılarının kurulması,
2. Giriş: Uygun döngü yapılarının kurulması,
3. Giriş: Algoritma ve akış şeması dallanmalarının doğru yapılması,
4. Giriş: Algoritma ve akış şeması tanımlamalarının doğru yapılması,
5. Giriş: (varsa) test genel başarı puan ortalaması.

Yapay Sinir Ağı'nın çıkış katmanında ise zorluk düzeyi ve dönüt değeri bilgileri elde edilebilmektedir. Buna göre; her bir öğrenci, gerçekleştirdiği aktiviteler sonrasında belirli zorluk düzeyi ve dönüt değeri bilgilerine sahip olmaktadır. Sistem böylece yeni problemleri görüntülerken öğrencilerin zorluk düzeyi ve dönüt değeri bilgileri dikkate alarak seçim işlemi gerçekleştirmektedir.

Kurulan Yapay Sinir Ağı modelinin temsili bir şeması Şekil 1 altında sunulmuştur. Metni teknik detaylara boğmamak adına Yapay Sinir Ağı tekniğinin detaylarına değinilmemiştir. Bu nedenle ilgili tekniğe ilişkin bilgi edinmek isteyen okurlara genel kaynak olarak (Anderson & McNeill, 1992; Basheer & Hajmeer, 2000; Hassoun, 1995; Yegnanarayana, 2009) önerilmektedir.



Şekil 1. Akıllı yazılım sisteminde kullanılan Yapay Sinir Ağı modeline yönelik şema.

Yapay Zekâ mekanizmasının işleyişine yönelik olarak ifade edilmesi gereken diğer bazı noktalar da şu şekildedir:

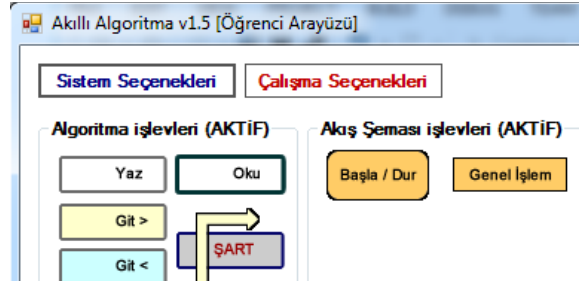
- Sistemde her yeni algoritma / akış şeması problemi tanımlama sonrası örnek giriş puanları (varsa testler de eklenerek) ve beklenen zorluk düzeyi ile dönüt değeri tanımlamaları yapılarak Yapay Sinir Ağı sisteminin kendi kendini eğitmesi sağlanmaktadır.
- Böylelikle sistemde her yeni problem aktivitesi yerine getirildikten sonra öğrencinin elde ettiği zorluk düzeyi ve dönüt değeri bilgilerine göre iki mekanizma devreye girmektedir. Buna göre:
 - o Sistem bir sonraki örnek problemi elde edilen zorluk değerine en yakın durumdaki rastgele adaylardan seçmektedir.
 - o Elde edilen dönüt değerinin ağırlığına göre puan kıstaslarına yönelik verilmiş olan dönütler işletilmektedir.

Genel Kullanım Özellikleri

Yazılım sisteminde öğretmenler ve öğrenciler için iki ayrı arayüz yapısı vardır. Öğreticilere yönelik arayüzler daha çok problem, muhtemel problem çözümleri, dönütler, testler... vb. unsurları tanımlamaya, öğrenci durumlarını ön izlemeye yönelik iken; öğrenci arayüzleri problem ve test çözmeye yöneliktir. Sistemin en önemli özelliği, günümüzde oldukça revaçta olan sürükle - bırak mekanizmasına dayalı problem tanımlama / çözme özelliklerine ve işlevlerine sahip olmasıdır.

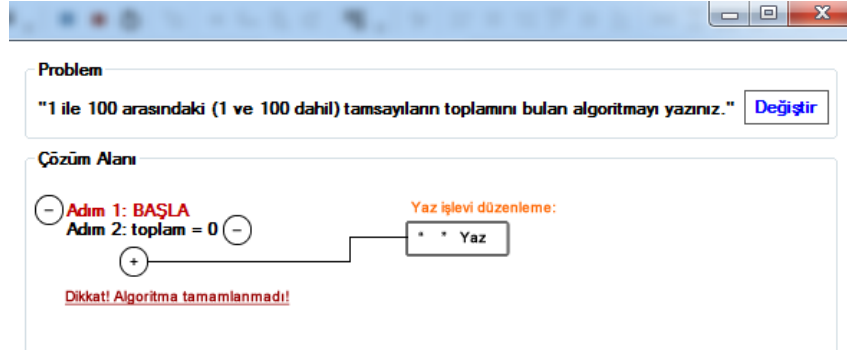
Yazılım sistemi, hızlı ve pratik bir kullanım tecrübesi sağlamak için Windows Form yapısında tasarlanmış ve C# programlama diliyle Visual Studio 2012 geliştirme ortamında kodlanmıştır. Görsel tasarımda gereksiz ayrıntılardan sakınılmış, sadece öğretim / öğrenim sürecine katkı sağlayacak unsurların işe koşulması amaçlanmıştır.

Yazılım kapsamında tipik bir öğrenci arayüzü, sol alanda sürükle - bırak mekanizmasıyla kullanılacak algoritma ve akış şeması kontrolleri, sağ alanda ise aktif probleme yönelik unsurlardan ve çözüm alanından oluşmaktadır. Yine arayüzün üst kısmında yazılıma ilişkin çeşitli işlevlere ulaşılabilir. Algoritma ve akış şeması kontrollerine yönelik bir ekran görüntüsü Resim 1'de sunulmuştur.



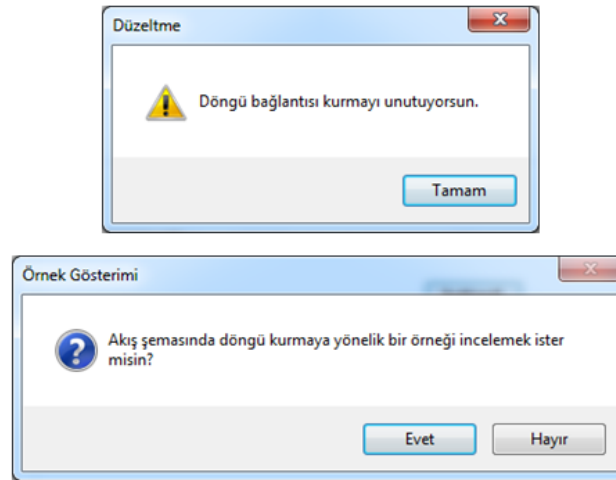
Resim 1. Öğrenci arayüzünde algoritma ve akış şeması kontrolleri.

Çözüm alanında oluşturulan her algoritma / akış şeması yapısı, ilgili kontroller üzerine tıklanarak düzenlenebilmekte ve muhtemel bir çözüm tamamlandıktan sonra sistem tarafından değerlendirilebilmesi için öğrenci tarafından onaylanabilmektedir. Resim 2 kapsamında örnek bir problem ve çözüm alanından ekran görüntüsü görülebilmektedir.



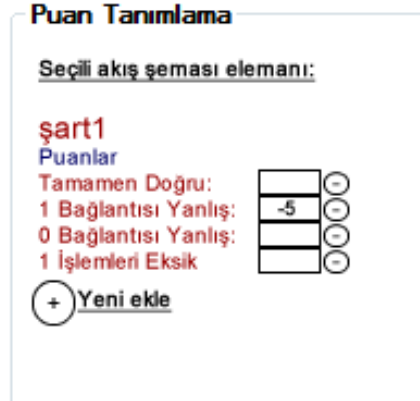
Resim 2. Örnek bir problem ve çözüm alanı.

Yapay Zekâ mekanizması açıklanırken de değinildiği üzere, öğrencilerin sahip oldukları zorluk düzeyi ve dönüt değeri bilgileri, yeni - uygun problemlerin seçilmesine ve problemler esnasında yazılım sistemi tarafından dönütler sağlanmasına ön ayak olmaktadır. Bu noktada ilgili dönütler, yazılım tarafından öğrencilerin farklı problemlere veya kaynaklara yönlendirilmesini veya problem çözümlerinde yapılan hatalar konusunda uyarılarda bulunulmasını amaçlamaktadır. Resim 3 altında bazı örnek dönütler gösterilmiştir.



Resim 3. Yazılım sisteminden örnek dönütler.

Öğretici arayüzü ele alındığında ise, daha çok algoritma / akış şeması problemlerinin veya genel materyallerin, testlerin... vb. yan unsurların oluşturulabildiği bir arayüz ile karşı karşıya kalınmaktadır. Özellikle algoritma / akış şeması problemleri oluşturulurken öğrenciler problem tanımlamaları ve her bir problem için muhtemel çözüm senaryolarını oluşturabilmektedir. Çözüm senaryoları oluşturulurken, - tıpkı öğrenci arayüzünde olduğu gibi- algoritma veya akış şeması kontrolleri sürükle - bırak yaklaşımı ile bir araya getirilebilmektedir. Eklenen her yeni kontrolün, öğrenci tarafından yapılan işlemlere veya oluşturulan bağlantılara göre puanlandırılması sağlanabilmektedir. Seçili bir akış şeması kontrolüne yönelik olarak ekrana getirilen puanlama panelinden örnek bir ekran görüntüsü Resim 4 altında sunulmuştur.



Resim 4. Seçili bir akış şeması kontrolüne yönelik olarak ekrana getirilen puanlama paneli.

Kısaca açıklanan arayüzler sayesinde, hem öğrencilerin hem de öğrencilerin etkileşimli bir eğitimsel yazılım ortamında aktif duruma gelmesi; böylece algoritma ve akış şeması kavramlarına yönelik bilgi ve becerilerin kazandırılmasına dayalı aktiviteleri yerine getirmesi sağlanabilmektedir. Böylelikle spesifik programlama dilleri kapsamında bilgisayar programlamaya yönelik ileri düzey konulara geçmeden önce, algoritmik düşünce mantığının istendik düzeylerde tecrübe edinilmesi (ve dolayısıyla kazanılması) mümkün olmaktadır.

Her ne kadar çalışmanın amaçlarına uygun olarak tasarlanıp geliştirilmiş olsa da, akıllı eğitimsel yazılım sisteminin etkinliğini ve başarımını bir şekilde değerlendirmek gerekmektedir. Bu nedenle sonraki bölümde, gerçekleştirilen değerlendirme süreçleri açıklanmıştır.

Yöntem

Yapay Zekâ mekanizması ve genel kullanım özellikleri açıklanan akıllı eğitimsel yazılım sistemi ve bu sistem ile uygulanan süreç, istendik etkinliğin ve başarımın sağlanıp sağlanmadığını anlamak adına tipik bir değerlendirme sürecinden geçirilmiştir. Sürece ilişkin detaylar bu bölümde, ilgili alt-başlıklar altında sunulmaktadır.

Değerlendirme Yaklaşımı

Geliştirilen yazılımın, algoritma ve akış şemaları kavramları öğretiminde ne derece etkin ve başarılı olduğu sorusu, kuşkusuz ki öğretim sürecine tabi olan öğrenciler tarafından daha iyi cevaplandırılacaktır. Bu bağlamda, yazılımı kullanarak edinilen tecrübelerin ve kullanım sonrası ortaya çıkan birtakım verilerin, yazılımın değerlendirilmesi noktasında kullanılabileceği düşünülmüştür. Bu yüzden; etkin değerlendirmeyi sağlamak adına, ilgili yazılım sisteminin bilgisayar programlama temellerine yönelik dersler alan öğrenciler tarafından kullanılması sağlanmış ve genel hatlarıyla niceliksel bir değerlendirme / araştırma yaklaşımı tercih edilmiştir. Yaklaşım kapsamında ilk olarak deneysel araştırma modeli

üzerinde durulmuş, beraberinde ise sistem üzerinde sunulan problemlerin öğrenciler tarafından uygunluğu / kabul derecesi ölçülmüştür. Yazılımın kullanıldığı süreç sonunda, -yine süreci tamamlamak adına- yazılımı kullanmış olan öğrencilerin görüşleri bir anket çalışması yardımıyla dikkate alınmıştır. Sürecin dengelerini daha iyi anlamak adına deneysel sürece tabi olan deney ve kontrol gruplarından kısaca bahsetmek gerekmektedir.

Çalışma Grubu - Deney ve Kontrol Grupları

Değerlendirme sürecine, Uşak Üniversitesi'nde Programlama Temelleri ve Bilgisayar Programlama gibi, bilgisayar programlamaya giriş niteliğinde dersler alan toplam 90 ön lisans öğrencisi katılmıştır. Öğrencilerden 45 tanesi, akıllı yazılım sisteminin kullanıldığı deney grubunu, geriye kalan 45 tanesi ise geleneksel eğitim sürecinin izlendiği kontrol grubunu oluşturmuştur. Bu noktada, uygulanan deneysel değerlendirme sürecinin bir yarıyıl (14 hafta) boyunca devam etmesi sağlanmıştır. İşlenen derslerde aynı müfredatın, benzer dengelerdeki öğretim elemanları tarafından işlenmesine dikkat edilmiş ve yine grupların oluşturulması esnasında, dersleri alan bütün öğrencilerin genel başarı düzeyleri ve derslerdeki ilgileri - tutumları dikkate alınarak; dengeli bir yapının ortaya konulmasına çalışılmıştır. Değerlendirmeye gölge düşürebilecek çevresel faktörler süreç öncesi kontrol altına alınmıştır.

Değerlendirme Araçları ve Verilerin Toplanması

Anlaşılabileceği üzere; değerlendirme sürecinden elde edilecek veriler, uygulanan deneysel süreç esnasına ve sonrasına tekabül etmektedir. Bu noktada, yazılım sisteminin kullanımının öğrenci akademik başarısını nasıl etkilediğini anlamak için yarıyıl sonunda ilgili dersleri alan deney ve kontrol grubu öğrencilerinin durumları dikkate alınmıştır. Değerlendirme süreci esnasına yönelik diğer değerlendirme aracı ve veri toplama yaklaşımında ise, yazılım sistemi ortamında deney grubu öğrencilerine sunulan her bir problemin öğrencilerin düzeylerine uygunluk durumunun (eksik olunan konulara isabet, uygun dönütte bulunma, eksik olunan konulara yönlendirme... vb. durumlara göre) 5 yıldızlı bir ölçekte oylanması istenmiştir. Son olarak, süreç sonrasında, yine deney grubu öğrencilerine sunulan anket çalışmasıyla yazılıma ve sürece yönelik görüşlerin elde edilmesi sağlanmıştır. Toplam 24 maddeden oluşan ankette dönütler 5'li Likert Ölçeği (1: Kesinlikle Katılmıyorum; 2: Katılmıyorum; 3: Kararsızım; 4: Katılıyorum; 5: Kesinlikle Katılıyorum) kapsamında elde edilmiştir. Uzman görüşü eşliğinde hazırlanmış olan bu ifadelerin yanı sıra, öğrencilerin anketteki ilgili alana yazdıkları kişisel düşünceler ve öneriler de toplanmıştır.

Verilerin Analizi

Değerlendirme süreci kapsamında uygulanan deneysel yaklaşımına göre; yarıyıl sonunda ilgili dersleri geçen deney ve kontrol grubu öğrenci sayıları ve başarı notu ortalamaları gibi veriler, yazılımın akademik başarıya etkisini anlamak adına dikkate alınmıştır. Bu bağlamda, her bir öğrencinin başarı notu, yarıyıl ortasında yapılan vize sınavının %40'ı ve yarıyıl sonunda yapılan final sınavının %60'ı toplanarak hesaplanmıştır (Her iki sınav da test ve uygulamaya dönük klasik sorulardan oluşmaktadır). Diğer yandan, sunulan problemlerin uygunluğuna yönelik verilen (5 yıldızlı ölçekteki) oylar, yazılımın Yapay Zekâ mekanizmasının başarımı doğrultusunda fikir edinilmesi için kullanılmıştır. Süreç sonrası anket çalışmasında ise; her bir maddeye verilen dönütler ve kişisel düşünce - öneriler dikkate alınmış ve bu bulgular da, gerek yazılım sisteminin, gerekse eğitim sürecinin başarısını anlamada söz sahibi olmuştur.

Bulgular

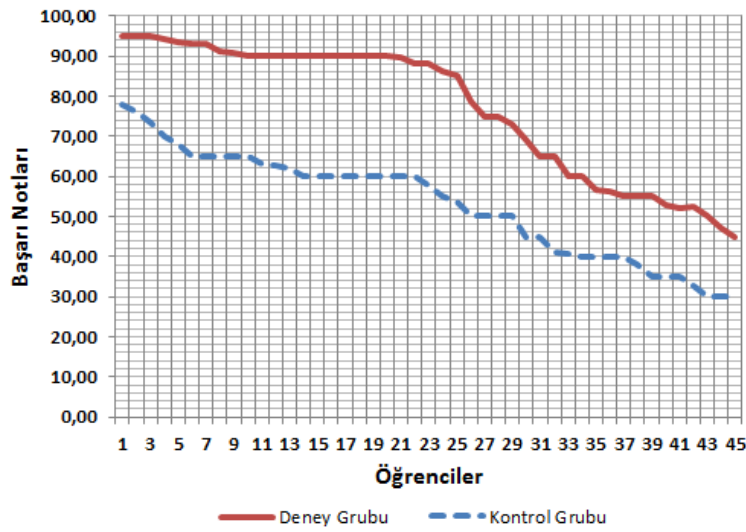
Deneysel değerlendirme yaklaşımı kapsamında, ilgili yarıyıl sonunda elde edilen akademik başarımların bulgularını, her iki grup için Tablo 1 altında sunmak mümkündür.

Tablo 1.

Deneysel değerlendirme yaklaşımı bulguları.

Grup	Toplam Öğrenci Sayısı	Dersi Geçen Öğrenci Sayısı	Ortalama	Standart Sapma
Deney Grubu	45	34	77.05	16.90
Kontrol Grubu	45	22	52.93	13.38

Tablo 1'den görüleceği üzere, deney grubu kontrol grubuna göre daha yüksek bir akademik başarımlar göstermiştir. Her iki gruba yönelik başarımların dağılımları grafiksel olarak Şekil 2'de gösterilmiştir.



Şekil 2. Deney ve kontrol gruplarının başarımlarının dağılımları.

Süreç sırasında sunulan problemlerin uygunluk durumuna yönelik olarak, yazılım sistemini kullanan deney grubu öğrencilerinin vermiş olduğu oylara yönelik bulgular Tablo 2 altında sunulmuştur.

Tablo 2.

Problemlerin uygunluk durumlarına yönelik verilen oylar kapsamında bulgular.

Problem Kategorisi	Toplam Problem	Toplam Oy ^a	Yıldız Sayısına Göre Oylar					Ortalama
			1	2	3	4	5	
Genel İşlemler	14	630	0	10	39	189	392	4.53
Şart - Koşul Kontrolleri	17	765	0	0	57	214	494	4.57
İç İçe Şart - Koşul Kontrolleri	11	495	0	3	61	185	246	4.36
Döngüler	22	990	10	13	72	264	631	4.51
İç İçe Döngüler	15	675	4	5	55	203	408	4.49
Matematiksel İşlemler	12	540	0	0	25	151	364	4.63
Öz Yinelemeli İşlemler	6	270	1	2	12	31	224	4.76
Alt İşlev Kullanımı	10	450	0	0	6	54	390	4.85
Özel İşlev Kullanımı	7	315	0	0	11	93	211	4.63

^aToplam Oy = Problem Sayısı * 45 (Deney Grubu Öğrenci Sayısı)

Tablo 2’de, yazılım sistemi ortamında sunulan toplam 114 adet algoritma / akış şeması problemi baskın / hedef öğretim yönlerine göre 9 ayrı kategori altında sunulmuş ve elde edilen farklı oy sayıları, ortalama değerler ile birlikte verilmiştir. Bulgulardan anlaşılacağı üzere, genel anlamda deney grubu öğrencilerinin sunulan problemleri uygun bulduğunu ifade etmek mümkündür.

Eğitimsel süreç sonrası deney grubuna uygulanan anket çalışması, maddeleri ve her madde için elde edilen dönüt sayılarına (bulgulara) göre, Tablo 3 kapsamında verilmiştir.

Tablo 3.
Anket çalışması ve elde edilen bulgular.

No	İfade	Dönütler ^a				
		1	2	3	4	5
1	“Bu eğitim süreci öğrenim tecrübelerimi olumlu yönde etkiledi.”	0	1	1	11	32
2	“Bu eğitim süreci algoritmik düşünce yaklaşımımı geliştirdi.”	0	1	4	12	28
3	“Bu süreç sayesinde soyut ve anlaşılması zor kavramları öğrendim.”	0	0	4	11	30
4	“Buna benzer bir eğitimsel süreçte tekrar yer almak istemiyorum.”	27	15	3	0	0
5	“Eğitim süreci dersteki akademik başarımla düzeyimi artırdı.”	0	1	3	12	29
6	“Farklı öğretim / öğrenme yaklaşımlarını daha çok tercih ederim.”	31	10	4	0	0
7	“Bu süreçte bir yazılım sisteminin yer almasını doğru bulmuyorum.”	30	9	3	2	1
8	“Bu eğitim sürecini başka arkadaşlarımla da tecrübe etmesini isterim.”	0	0	3	9	33
9	“Dersteki başarı düzeyimde bu süreç birinci derece etkili olmuştur.”	0	1	1	9	34
10	“Eğitim sürecindeki aktiviteler beni sıktı.”	26	12	5	2	0
11	“Bu süreç sayesinde etkili ve verimli bir kendi kendine öğrenme yaşadım.”	1	1	4	8	31
12	“Bu eğitim süreci dersin amaçlarına ve kapsamına uygun değildir.”	29	14	2	0	0
13	“Yazılım sistemi etkin ve verimli bir öğrenme süreci sağlıyor.”	0	1	1	10	33
14	“Yazılım sistemini kolay ve hızlı bir şekilde kullanabildim.”	2	2	4	11	26
15	“Bu yazılım sisteminin geliştirilmesini isterim.”	0	0	2	6	37
16	“Bu yazılım sistemi ile bilgisayar programlama öğretilir.”	0	1	2	12	30
17	“Başka programlama derslerinde de bu yazılımın kullanılmasını isterim.”	0	0	2	10	33
18	“Bu yazılım sistemi ile algoritma kavramını daha iyi anladım.”	0	2	4	8	31
19	“Bu yazılım sistemi ile akış şeması kavramını daha iyi anladım.”	0	0	2	9	34
20	“Bu yazılım sistemi ile iyi bir öğrenme sağlanamaz.”	32	10	2	1	0
21	“Yazılım sistemini kullanmak kafamı karıştırdı.”	27	8	10	0	0
22	“Bu yazılım ile şart – koşul, döngü gibi kavramları daha kolay öğrendim.”	0	3	4	8	30
23	“Yazılım sisteminin sağladığı dönütler öğrenme sürecimi olumlu etkiledi.”	1	1	2	9	32
24	“Yazılım sisteminin tasarımı ilgi çekici değil.”	35	8	2	0	0

^a5’li Likert Ölçeği (1: Kesinlikle Katılmıyorum; 2: Katılmıyorum; 3: Kararsızım; 4: Katılıyorum; 5: Kesinlikle Katılıyorum)

Anket çalışmasıyla elde edilen bulgular, yazılım sistemini tecrübe eden öğrencilerin, maddelere genel olarak olumlu yönde dönütlerde bulunduğunu göstermiştir.

Anket bulgularına ek olarak; deney grubu öğrencilerinin yazılım sistemine ve sürece ilişkin düşünceleri ve önerileri de dikkate alınmıştır. Buna göre, öğrencilerden gelen dikkat çekici bazı düşünce ve önerileri şu şekilde sıralamak mümkündür:

- “Kullandığımız program olmasaydı bu derste çok zorlanırdım.”
- “Bu programı diğer programlama derslerinde de kullanalım.”
- “Program dersler sonrası konulara daha kolay çalışmamı sağladı.”
- “İlk kez oyunlar dışında bilgisayar başına bu kadar bağlandım.”
- “Programı telefonda da kullanmak isterdim.”

- “Programı sevdim. Hocam Antep’ten isteğiniz olursa, her zaman.”
- “Ders BA düştü AA düşün istiyordum ama program iyiydi hocam.”
- “Diğer programlama derslerinde de buna benzer bir dönem geçirmek istiyorum.”
- “Verimli bir ders dönemi geçirdim.”
- “Geçen sene kaldığım dersin aslında kolay olduğunu anladım.”

Tartışma

Gerçekleştirilen değerlendirme süreci sonucunda elde edilen bulgulara göre, geliştirilen yazılım sistemine ve uygulanan eğitimsel sürece yönelik şu açıklamaları yapmak mümkündür:

- Yarıyıl sonunda deney ve kontrol grubu arasında ortaya çıkan fark, akıllı eğitimsel yazılım sisteminin kullanımı sonucunda öğrencilerin akademik başarı düzeyinin artırılması yönünde pozitif gelişmeler elde edildiğini göstermektedir.
- Yazılım sistemini kullanan öğrencilere uygulanan anket çalışması ile elde edilen bulgular, öğrencilerin geliştirilen yazılım sisteminden ve tecrübe edinilen eğitimsel süreçten memnun olduğunu göstermektedir. Öğrenciler, benzeri süreçlerin ve yazılımın farklı bilgisayar programlama derslerinde de işe koşulmasını istemektedir.
- Bilgisayar programlama temellerine yönelik olarak anlaşılması zor olan soyut ve mantıksal konuların, akıllı yazılım sistemi sayesinde daha iyi öğrenildiği (bir bakıma öğretildiği) sonucuna ulaşılabilmektedir.
- Akademik başarı yönünden artan bir ivme kazanılması ve bilgisayar programlama gibi soyut bilgilerin pratik platforma taşınmasının önem arz ettiği bir öğrenim sürecinde, etkin ve sıkıcı olmayan ders aktivitelerinin elde edilmesi, başarılı bir öğretim yaklaşımı ve bu yönde şekillenmiş bir sistemin işe koşulduğunu göstermektedir.
- Yazılım sistemi bağlamında kullanılan Yapay Zekâ tabanlı mekanizmanın, yazılım desteğiyle gerçekleşen öğretim sürecini benzer eğitsel yazılım yaklaşımlarına göre bir adım öteye taşıdığı sonucuna ulaşmak mümkündür.
- Deney grubu öğrencilerinin yazılım sistemi tarafından sunulan problemlerin uygunluk durumları yönünde verdikleri oylar, Yapay Zekâ mekanizmasının isabetli ve istikrarlı bir problem seçim / sunum yaklaşımı izleyebildiğini göstermektedir.
- Yapay Zekâ mekanizmasının başarımı, Yapay Zekâ destekli yazılım kullanımının eğitimsel süreçlere sağladığı olumlu etkileri gösteren literatürdeki çalışmaların desteklenmesi noktasında önemlidir.
- İlerleyen süreçlerde verilen yeni bilgisayar programlama derslerinde de görülmüştür ki; algoritma ve akış şeması kavramlarına yönelik bilgi ve becerilerini sağlam temellere oturtmuş olan öğrenciler; farklı programlama dillerinin öğrenilmesi aşamasında daha az problem yaşamıştır.
- Deney grubu öğrencilerinin, yazılım sisteminin daha fazla derste ve platformda (örneğin mobil cihazlar) kullanılabilmesi yönündeki önerileri, çalışmanın başarımı ve gelecek süreçteki potansiyeli bağlamında olumlu sonuçlar ortaya koymaktadır.

Sonuçlar ve Gelecek Çalışmalar

Bu çalışmada, bilgisayar programlamanın temellerini teşkil eden algoritma ve akış şeması kavramlarını etkin bir şekilde öğretmeyi amaç edinen, akıllı bir eğitimsel yazılım sistemi incelenmiş ve hem yazılım sistemini, hem de yazılım sistemi ile gerçekleştirilen eğitimsel süreci değerlendirmek adına gerçekleştirilen çalışmalara odaklanılmıştır. Bu noktada, değerlendirme süreçleri ile ulaşılan bulgular ve dolayısıyla söz konusu yazılım sistemi ile uygulanan eğitimsel sürecin başarımı tartışılmıştır.

Değerlendirme süreçleri ile ulaşılan bulgular göstermektedir ki, geliştirilen akıllı yazılım sistemi, algoritma ve akış şeması kavramlarının öğretimi bağlamında etkili bir araç olarak kullanılabilir. Böylece bu yazılım sisteminin kullanıldığı eğitimsel süreçler sonrasında, -geleneksel yaklaşımların aksine- öğrencilerin ileri düzeyde bilgisayar programlama konularına daha hazır düzeyde olmaları sonucuna ulaşılabilir. Yazarlar bu bağlamda çalışmanın Yapay Zekâ tabanlı eğitimsel yazılımların uygulanması ve etkinliğinin değerlendirilmesi yönünde literatürdeki benzer çalışmalara alternatif olacağını düşünmektedir.

Elde edilen sonuçlar da dikkate alınarak, gelecek çalışmalar kapsamında şu düşüncelerden / planlardan bahsedilebilir:

- Yazılım sisteminin akıllı öğretim yapısı ek mekanizmalarla desteklenecektir.
- Yazılım sisteminin farklı koşullar altında değerlendirilerek başarımın değerlendirilmesine devam edilecektir.
- Yazılım sisteminin Web ve mobil tabanlı platformlar için de uyumlu hale getirilmesi sağlanacaktır.
- Yazılım sisteminin ileri düzeydeki diğer bilgisayar programlama derslerinde kullanılabilmesi için bazı programlama dillerine yönelik yeni özellik ve işlevlerin (hatta yeni, güncellenmiş bir Yapay Zekâ altyapısının) eklenmesi / geliştirilmesi sağlanacaktır.

Extended Abstract

Introduction

Computer programming is an essential ability, which should be owned in order to make computer based systems working according to needs and direct them to solve problems. As a result of rapid developments in today's technology, and also changes like reaching to the needed information fast, essential knowledge and abilities regarding to computer programming have become necessary for different research / science / activity fields. Even recent developments show that spread rate of computer programming approach has increased (Kafai & Burke, 2015; Sterritt et al., 2015). This increased interest to the computer programming has caused educational processes to be subjected to many works; in the sense of obtaining the related knowledge and abilities. Performed works have brought the experiences, which are milestone for teaching and learning computer programming, into the fore. It can be said that the most important experience in learning computer programming is the process of learning algorithm and flowchart concepts. Because of this, research works planned for teaching these concepts effectively have become popular recently.

When the related literature is examined, it can be seen that there are different problems that should be solved in point of teaching / learning computer programming. Especially encountering difficulties in effectively obtaining problem solution approaches and solution designing abilities may cause finding the ability of computer programming as a difficult approach to be learned (Moser, 1997; Pillay, 2003; Pillay & Jugoo, 2005). In order to overcome this problem, different research works in understanding computer programming ability, obtaining it via alternative methods and removing students' dropout rates have been performed (Bau et al., 2015; Bennedsen & Caspersen, 2007; Cañas et al., 1994; Cutts et al., 2014; Esteves et al., 2011; Krpan et al., 2015; Michaelson, 2015; Robins et al., 2003; Wiedenbeck et al., 2004; Wiedenbeck et al., 1999; Yadin, 2011). An important trend appeared as a result of the performed works is introducing teaching environment, which are easy-to-use, visually rich, and interactive. For example, Scratch, which was developed at the Massachusetts Institute of Technology, is used as an effective tool for algorithms and algorithmic think logic, which are also essential subjects of computer programming (Armoni et al., 2015; Maloney et al., 2010; Resnick et al., 2009). There are also different applications, which are using visual processes and focused on teaching computer programming and algorithmic thinking logic (Carlisle et al., 2005; Chen & Morris, 2005; Hooshyar et al., 2015). Another matter that should be expressed is that Artificial Intelligence approaches, methods, and techniques are widely used actively for designing and developing educational software systems. Ergo, Artificial Intelligence support, which allows employing intelligent organization, evaluation and usage features, is still in educators' interest (Dorça, 2015; Jain et al., 2013; Matsuda et al., 2015; Wenger, 2014).

As it is known, different software systems aiming to provide highly interactive and practical application processes at the time of teaching computer programming logic are often developed by researchers in especially recent years.

The software system introduced in the work is tightly connected to the rule of teaching essential computer programming concepts, but ensures this task in the context of an Artificial Intelligence supported, intelligent mechanism. At this point, objective of this work is "to introduce an Artificial Intelligence supported educational software system, which has been developed for teaching important subjects of computer programming: algorithm and flowchart concepts, and touch upon the findings, which were obtained for evaluating success of the system". Additionally, it is possible to mention sub-objectives as follows:

- Is there any difference in academic achievements regarding to the experimental group students, who have used the intelligent software system along their educational process and the control group students, who have experienced the traditional educational process.
- Can the appropriate ones from the problems over the system be provided intelligently to the experimental group students, who have used the intelligent software system?
- Do the experimental group students, who have used the intelligent software system, have positive ideas about using such software system in their courses?

Intelligent Software System in Teaching Algorithm and Flow Chart

It is possible to examine the intelligent software system introduced in this study from perspectives of Artificial Intelligence mechanism and using features – functions.

Artificial Intelligence Mechanism

The related intelligent software system employs an Artificial Neural Networks approach. This approach takes students' activity / problem solving points as inputs and determines difficulty level and feedback values for each of them. After that, the system chooses the next problem that will be viewed to the student according to these dynamic values. Readers are referred to (Anderson & McNeill, 1992; Basheer & Hajmeer, 2000; Hassoun, 1995; Yegnanarayana, 2009) in order to learn more about the Artificial Neural Networks.

General Using Features

In the software system, there are two separate interface structures for teachers and students. In the interfaces for teachers, it is possible to define problems, possible solutions, feedbacks...etc. and track students' status whereas student interfaces are for solving problems and tests. The most important feature of the system is that it is possible to use the drag and drop approach for all problem defining and solving tasks.

Method

Developed software system and the applied educational process via this system have been evaluated in order to have idea if desired effectiveness and success were achieved.

Evaluation Approach

The question of how the developed software is effective and successful at teaching algorithm and flowchart concepts can be answered better by the students, who have experienced the educational process. In this sense, it was thought that the usage experiences and the data obtained after the related educational process will make it easier to evaluate the software system. Because of this, students taking courses related to essentials of computer programming were enabled to use the software system and in this way, a quantitative evaluation / research approach has been chosen. In the context of this approach, an experimental research model has been applied and also appropriateness / acceptance rate of the provided problems over the system has been measured. After the software usage process, students' ideas about the software system and the experienced educational process have been taken into consideration, thanks to a survey work.

Working Group - Experimental and Control Groups

A total of 90 undergraduate students taking courses like Essentials of Computer Programming and Computer Programming at Usak University have taken active part in the evaluation process. At this point, 45 students have formed the experimental group (the group using the software system) while the remaining ones have formed the control group (the group experiencing traditional educational process). The experimental evaluation process has been performed along one term (14 weeks). In the related courses, same educational programs were used by the teachers at similar levels. During the formation of groups, students with similar academic achievement levels have been chosen in order to ensure a balanced experimental setup. The related environmental factors having the potential of affecting the process negatively have been taken under control.

Evaluation Tools and Data Gathering

In order to understand how employment of the software system affected students' academic achievements, situations regarding to students of experimental and control groups have been taken into consideration at the end of term. According to the other evaluation tool and the data gathering approach, students have been wanted to give their votes for the appropriateness of the provided problems (meeting with the incomplete course subjects, giving appropriate feedback, directing to appropriate subjects...etc.) in a 5-star scale. Finally, ideas regarding to experimental group students have been gathered by using the survey tool. For the survey having a total of 24 statements, students have been wanted to give their feedback in the sense of Likert Scale (1: Totally Disagree; 2: Disagree; 3: No Opinion; 4: Agree; 5: Totally Agree). In addition to these statements, which have been prepared via expert support, also personal ideas and suggestions written by the experimental group students have been taken into consideration.

Data Analysis

In the sense of the experimental evaluation approach, data like number of experimental and control group students passed the courses and also average success points have been taken into consideration. At this point, each student's success point has been calculated by summing 40% of the visa examination and also 60% of the final examination (Both two examinations include test and applied classical questions). On the other hand, votes regarding to appropriateness of the provided problems over the system have been used for having idea about success of the Artificial Intelligence mechanism. Thanks to the last evaluation approach: survey tool, students' feedback have been used for evaluating the success of the software system and the educational process.

Discussion

According to the obtained findings after evaluation process, it is possible to provide some explanations as follows:

- Difference between experimental and control group at the end of term shows that using intelligent educational software system had positive effects on improving the related students' academic achievements.
- Findings obtained via survey tool show that the related students were satisfied with the developed software system and the experienced educational process. Students have wanted the related educational process to be applied within different computer programming courses.
- It has been seen that difficult, abstract and logical subjects can be learned better, thanks to the intelligent software system.

- Having an improved academic achievement and also ensuring effective and not boring course activities for computer programming oriented learning process indicate a successfully formed teaching approach and system.
- It is possible to express that the Artificial Intelligence based mechanism takes the software supported teaching process one step away according to similar educational software approaches.
- Votes received by the students for evaluating appropriateness of the provided problems show that the Artificial Intelligence mechanism is able to ensure an accurate and effective approach in problem choosing / providing.
- Success of the Artificial Intelligence mechanism is important to support the related literature works, which show positive effects of using Artificial Intelligence supported software along educational processes.
- It has been also observed in the next computer programming courses that students, who had had necessary knowledge and ability level on algorithm and flowchart concepts, experienced less problems on learning different programming languages.
- Students' suggestions for using the software system in more courses and over different platforms (for example mobile devices) indicate positive results about success of the work and its potential in the future.

Conclusions and Future Work

In this work, an intelligent educational software system, which aims to effectively teach algorithm and flowchart concepts, have been examined and the related works performed for evaluating both software system and the performed educational process via this system have been reported. At this point, findings obtained via evaluation processes and the success of the educational process, which was performed thanks to the software system, have been discussed.

According to the findings obtained via evaluation process, developed intelligent software system can be used as an effective tool for teaching algorithm and flowchart concepts. Thus, it can be expressed that the students experienced educational processes supported with this software system will be able to focus on and learn advanced computer programming subjects. In this sense, the authors think that this work is an alternative for similar works, which are based on applications of Artificial Intelligence supported educational software systems and evaluating such tools.

Obtained results have encouraged the authors to plan / perform some future works. These works can be expressed briefly as follows:

- Intelligent teaching approach of the software system will be supported with additional mechanism.
- Evaluating the success of software system will continue in the future by performing new evaluation works; under different conditions.
- New versions of the software system will support Web and mobile platforms.
- In order to use the software system in other advanced computer programming courses, features and functions of some programming languages (even a new, updated Artificial Intelligence infrastructure) will be added / improved.

Kaynakça

- Anderson, D., & McNeill, G. (1992). Artificial neural networks technology. A DACS state-of-the-art report. Kaman Sciences Corporation, 258, 13502-462.
- Armoni, M., Meerbaum-Salant, O., & Ben-Ari, M. (2015). From scratch to “real” programming. *ACM Transactions on Computing Education (TOCE)*, 14(4), 25.
- Basheer, I. A., & Hajmeer, M. (2000). Artificial neural networks: fundamentals, computing, design, and application. *Journal of Microbiological Methods*, 43(1), 3-31.
- Bau, D., Dawson, M., & Bau, A. (2015). Using Pencil Code to Bridge the Gap between Visual and Text-Based Coding. *46th ACM Technical Symposium on Computer Science Education (706-706)*. ACM.
- Bennedsen, J., & Caspersen, M. E. (2007). Failure rates in introductory programming. *ACM SIGCSE Bulletin*, 39(2), 32-36.
- Cañas, J. J., Bajo, M. T., & Gonzalvo, P. (1994). Mental models and computer programming. *International Journal of Human-Computer Studies*, 40(5), 795-811.
- Carlisle, M. C., Wilson, T. A., Humphries, J. W., & Hadfield, S. M. (2005). RAPTOR: a visual programming environment for teaching algorithmic problem solving. *ACM SIGCSE Bulletin*, 37(1), 176-180, ACM.
- Chen, S., & Morris, S. (2005). Iconic programming for flowcharts, java, turing, etc. *ACM SIGCSE Bulletin*, 37(3), 104-107.
- Cutts, Q., Connor, R., Michaelson, G., & Donaldson, P. (2014). Code or (not code): separating formal and natural language in CS education. *9th Workshop in Primary and Secondary Computing Education (20-28)*. ACM.
- Dorça, F. (2015). Implementation and use of Simulated Students for Test and Validation of new Adaptive Educational Systems: a Practical Insight. *International Journal of Artificial Intelligence in Education*, 1-27.
- Esteves, M., Fonseca, B., Morgado, L., & Martins, P. (2011). Improving teaching and learning of computer programming through the use of the Second Life virtual world. *British Journal of Educational Technology*, 42(4), 624-637.
- Hassoun, M. H. (1995). *Fundamentals of artificial neural networks*. MIT Press.
- Hooshyar, D., Ahmad, R. B., Yousefi, M., Yusop, F. D., & Horng, S. J. (2015). A flowchart-based intelligent tutoring system for improving problem-solving skills of novice programmers. *Journal of Computer Assisted Learning*.
- Jain, G. P., Gurupur, V. P., & Faulkenberry, E. D. (2013). Artificial intelligence based student learning evaluation tool. *Global Engineering Education Conference (EDUCON), 2013 IEEE (751-756)*, IEEE.
- Kafai, Y. B., & Burke, Q. (2015). Computer programming goes back to school. *Education Week*, 61-65.
- Krpan, D., Mladenović, S., & Rosić, M. (2015). Undergraduate Programming Courses, Students' Perception and Success. *Procedia-Social and Behavioral Sciences*, 174, 3868-3872.
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, 10(4), 16.
- Matsuda, N., Cohen, W. W., & Koedinger, K. R. (2015). Teaching the teacher: tutoring SimStudent leads to more effective cognitive tutor authoring. *International Journal of Artificial Intelligence in Education*, 25(1), 1-34.
- Michaelson, G. (2015). Teaching Programming with Computational and Informational Thinking. *Journal of Pedagogic Development*, 5(1).
- Moser, R. (1997). A fantasy adventure game as a learning environment: why learning to program is so difficult and what can be done about it. *ACM SIGCSE Bulletin*, 29(3), 114-116, ACM.
- Pillay, N. (2003). Developing intelligent programming tutors for novice programmers. *ACM SIGCSE Bulletin*, 35(2), 78-82.

- Pillay, N. & Jugoo, V. (2005). An investigation into student characteristics affecting novice programming performance. *ACM SIGCSE Bulletin* 37: 107-110.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11), 60-67.
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137-172.
- Sterritt, R., Hanna, P., & Campbell, J. (2015). Reintroducing programming to the school environment. *9th International Technology, Education and Development Conference*, (2), IATED.
- Wenger, E. (2014). *Artificial intelligence and tutoring systems: computational and cognitive approaches to the communication of knowledge*. Morgan Kaufmann.
- Wiedenbeck, S., Labelle, D., & Kain, V. N. (2004). Factors affecting course outcomes in introductory programming. *16th Annual Workshop of the Psychology of Programming Interest Group* (97-109).
- Wiedenbeck, S., Ramalingam, V., Sarasamma, S., & Corritore, C. (1999). A comparison of the comprehension of object-oriented and procedural programs by novice programmers. *Interacting with Computers*, 11(3), 255-282.
- Yadin, A. (2011). Reducing the dropout rate in an introductory programming course. *ACM Inroads*, 2(4), 71-76.
- Yegnanarayana, B. (2009). *Artificial neural networks*. PHI Learning Pvt. Ltd.

