

Mobil Programlama Eğitiminde Yeni Yaklaşımlar: App Inventor Örneği

Fatih Özding¹

¹ Selçuk Üniversitesi, Konya.
fatihozding@gmail.com

Özet: İnsanlar kendi ilgi ve yeteneklerine uygun mobil uygulamaları nasıl geliştireceklerini araştırmakta ve mobil programlama öğrenmeye yönlendirmektedir. App Inventor, mobil uygulama geliştirmek için görsel blokların kullanıldığı bir ortamdır. Bu ortamda, programlama bilgisini en az kullanarak öğrenenlerin kendi mobil uygulamalarını geliştirebilmeleri olanağı sunulmaktadır. AI, başlangıç düzeydekilerin prototip oluşturma sürecinde iyi bir seçenektir. Anlaması kolay olup kısa sürede uygulama oluşturma olanağı verir.

Anahtar Sözcükler: Programlama, mobil programlama, App Inventor

New Approaches in Mobile Programming Education: App Inventor

Abstract: People investigate how to develop their own mobile applications appropriate their interests and abilities and they are directed to learning mobile programming. App Inventor is an environment in which to develop mobile applications using visual blocks. In this environment, learners are offered the possibility to develop their own mobile applications by using minimal programming knowledge. AI is a good option at starting level in the process of creating prototypes. It is easy to understand and gives you the ability to create applications in a short time.

Keywords: Programming, mobil programming, App Inventor

1. Giriş

Bireylerin ilgi ve gereksinimlerine yönelik mobil uygulama sayısının her geçen gün artması, mobil araçların hayatın her alanında kullanımı yaygınlaşmaktadır. 2014 yılı Temmuz ayında mobil uygulama marketlerinde toplam 3 milyonu aşkın uygulama bulunmaktadır ve bunun 1 milyon 300 bin kadarını Android uygulamaları kapsamaktadır [18]. İnsanlar kendi ilgi ve yeteneklerine uygun mobil uygulamaları nasıl geliştireceklerini araştırmakta ve kendi basit mobil uygulamalarını oluşturacakları hazır ortamlara ya da mobil programlama öğrenmeye yönlendirmektedir. Farklı yaşlarda ve eğitim düzeylerine sahip kişilerin bu ilgi ve çabası, programlama eğitimini, okullarda verilen bir ders olmak dışında bir yaşam boyu öğrenme alanı olarak karşımıza çıkarmaktadır. Mobil teknolojiler, birçok mühendislik disiplinindeki, programlama yöntemlerinde yeni bir çağ başlatmıştır [19]. Mobil programlama sürecinde Android ortamı için Java, IOS ortamı için Objective-C, Windows Mobil ortamı için C# programlama dillerini öğrenmeye gereksinim vardır. Genel olarak bir Android programlama dersi iki aşamadan (iki dönem) oluşmaktadır. Birinci aşama Java programlamanın öğrenilmesini gerektirirken ikinci aşama uygulama (app) geliştirmeyi kapsamaktadır [19]. Ancak geleneksel programlama dillerinin karmaşık sözdizimi yapılarını, yeni

başlayan programcılar bir engel olarak görmektedir [6].

2. Programlama Eğitiminde Görsel Araçların Kullanımı

Sayısal/bilgisayarca düşünme (computational thinking), öğrencilere kazandırılması gereken okuma, yazma ve aritmetiğin devamındaki dördüncü analitik yetenek olmalıdır [20]. Sayısal/bilgisayarca düşünmenin erken yaştan itibaren ve programlama bilmeyen bireylere kazandırılması gerektiğine dair yaygın bir kabul olmasına rağmen [2,7], bunun öğretilmesi çabaları henüz yeni gelişmektedir [9]. Bu becerinin kazandırılması sürecinde programlama eğitimi önemli bir rol oynamaktadır. Ancak programlama, öğrenilmesi zor bir beceridir [10,14]. Bunun bir sonucu olarak, bilgisayar bilimi alanında giriş düzeyinde öğrenim gören öğrenciler tarafından programlama dersi zor olarak algılanan bir derttir [1].

Programlamada uzman olmak, uzun ve çaba gerektiren bir süreçtir. Programlama öğrenme sürecinde öğrenciler çeşitli sorunlarla karşılaşmakta ve süreçte başarısız olabilmektedir. Öğrenciler, programlamadaki başarısızlıklarının en büyük nedenlerini pratik eksikliği, algoritma oluşturmama ve bilgi eksikliği olduğunu belirtmektedir

[13]. Bilgi düzeyi yüksek olsa bile, programlama sürecinde ayrıntıları düşünme, dikkatsizlik, hata yapmaya neden olabilmektedir [12]. Özellikle programlamaya yeni başlayanlar için, programlama eğitimi sürecinin sürdürülebilir olması açısından, öğrenenleri bu hataları yapmasını en aza indirecek uygulamalara gereksinim vardır.

Programlamaya yeni başlayanlar yorumlama ya da derleme süreçleri hakkında yeterli düzeyde düşünemezler. Programın bilgisayar tarafından nasıl anlaşılacağı yeni başlayanlar için soyut bir kavramdır [6,14]. Programlamayı yeni öğrenenler, döngü yapıları [15], koşul yapıları [4] ve temel öğeleri kullanma dışındaki programı yapılandırma süreçlerinde [17] sorun yaşamaktadır. Doğal bir dili öğrenmek karmaşıktır. Programlama dilleri doğal karmaşıklığı yansıttığı için, matematiksel ve mantıksal belirginlik yükseldikçe, yeni başlayanların anlaması kolaylaşacaktır [6].

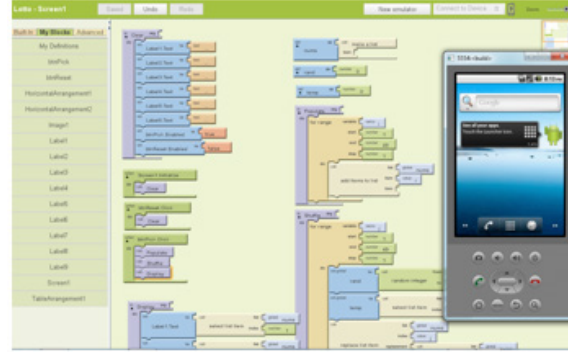
Bu zorluğu azaltmak için birçok bilgisayar eğitimcisi, öğrencilerin programlamaya giriş başarı olasılığını yükseltmek için yazılım araçları geliştirmeye yönelmiştir [5]. Öğrenciler temel programlamayı, kısa bir program yazmak yerine bir robot programlayarak, bir hikaye canlandırarak daha kolay öğrenebilirler [23]. Robot programlama ve programlamada çoklu ortam kullanımı, programlama öğrenenleri [21]. Geleneksel sözdizimleri yerine görsellerin kullanıldığı blok dilleri kullanımı, programlama eğitiminde son yıllarda yaygınlaşmaktadır. Blok dillerinde programlar şekilli görsel parçaların bir yap-boz gibi bağlanmasıyla oluşturulur [22]. Alice, BlueJ, Jeliot, Lego Mindstorms, RAPTOR ve Scratch gibi görsel ve blokların kullanıldığı görsel programlama dilleri kullanımı yaygınlaşmaktadır.

3. App Inventor

App Inventor (AI), programlamayı, yeni öğrenenler için eğlenceli ve ulaşılabilir yapmayı amaçlamaktadır. Günümüzün yeni nesil öğrencileri arasında, mobil araçların popülerliği ve aynı anda her yerde bulunma (ubiquity) özelliğinden dolayı, App Inventor'un öğrencilere programlama ve sayısal/bilgisayarca düşünme (computational thinking) kazandırma potansiyeli taşımaktadır. App Inventor, öğrencilerin, programlamanın ilgi çekici süreçlerine odaklanmalarına, problem çözmelerine ve arkadaşlarıyla paylaştıkları zaman, onlara sosyal olarak faydalı olacak yaratıcı mobil uygulamalar geliştirmelerine olanak sunar [11].

App Inventor, mobil uygulama geliştirmek için görsel blokların kullanıldığı bir ortamdır. Bu ortamda, programlama bilgisini en az kullanarak öğrenenlerin kendi mobil uygulamalarını geliştirebilmeleri olanağı sunulmaktadır. App Inventor'da blokların ilgili blok menüsünden seçilebilmesi,

blokların nasıl birleşeceğine yönelik şekillerin yönlendirici olması gibi özellikler, programlamaya yeni başlayanların öğrenme sürecinde sözdizimsel ve bilişsel başarısızlıklarını en aza düşürür. AI'nın kurulum gereksinimleri en az düzeyde olup tamamen tarayıcı tabanlıdır.



Şekil 1. App Inventor blok arayüzü

İlköğretimdeki öğrencilere programlama öğretmek için geliştirilen Scratch ile benzer blok dilini kullanmaktadır. Scratch'in devamı niteliğinde bir proje olan AI, Google tarafından geliştirilmiş daha sonra MIT Mobil Öğrenme Merkezi'ne aktarılmıştır. MIT, App Inventor'un ikinci sürümünü 2013 yılında geliştiricilerin hizmetine sunmuştur. App Inventor'un sadece çocuklar için değil aynı zamanda üniversitelerde verilen programlama derslerinin ilk aşamalarında başarılı olduğu görülmektedir [8]. AI, ABD'de okul sonrası programlarda, yaz kamplarında, öğretmen çalıştaylarında ve ortaokuldan üniversiteye kadar tüm bilgisayar derslerinde başarılı bir şekilde kullanılmaktadır [3].

AI ile programlama yaparken kullanılabileceğiniz yapılar, fonksiyonlar ve özellikler kısıtlıdır. Yalnızca "Tasarım Editörü" tarafından sağlanan fonksiyonlar ve "Blok Editörü"nde sağlanan bloklar bulunmaktadır. Örneğin dizi kullanımı bulunmamaktadır. Bu yüzden, AI, başlangıç düzeydekilerin prototip oluşturma sürecinde iyi bir seçenektir. Anlaması kolay olup kısa sürede uygulama oluşturma olanağı verir. Büyük ölçekli projeler için uygun değildir. Çünkü grafiksel programlama olanakları kısıtlamaktadır ve deneyimli programcı kod yazmada, blokları sürükleyip bırakmaya göre daha hızlı olacaktır.

App Inventor'un yaz okullarında ve programlama kamplarında kullanılması, ortaokullarda verilen BT ve Yazılım Dersi'ne eklenmesi, üniversitelerde programlama derslerinin ilk aşamalarında gösterilmesi öğrenenlerin programlama sürecindeki motivasyonları ve deneyimlerini sürdürmeleri üzerinde etkisi olacağı düşünülmektedir.

4. Kaynaklar

- [1] Aşkar, P. & Davenport, D., "An Investigation Of Factors Related To Self-Efficacy For Java Programming Among Engineering Students", The Turkish Online Journal of Educational Technology – TOJET 8(1), (2009).
- [2] Fletcher, G.H.L. & Lu, J.J., "Human Computing Skills: Rethinking The K-12 Experience", Communications of the ACM, 52(2), 23-25, (2009).
- [3] Gray, J., Abelson, H., Wolber, D. & Friend M., "Teaching CS principles with app inventor.", In Proceedings of the 50th Annual Southeast Regional Conference (ACM-SE '12). ACM, New York, NY, USA, 2012.
- [4] Green, T.R.G., "Conditional program statements and comprehensibility to professional programmers.", Journal of Occupational Psychology, 50:93-109 (1977).
- [5] Gross, P., & Powers, K., "Evaluating assessments of novice programming environments", Proceedings Of The First International Workshop On Computing Education Research, 99-110, Seattle, WA, USA, 2005.
- [6] Guzdial, M., "Programming environments for novices.", Computer Science Education Research. S. Fincher and M. Petre (Eds.). Swets and Zeitlinger. (2004).
- [7] Guzdial, M., "Paving the Way for Computational Thinking", Communications of the ACM, 51, 8, 27, (2008)
- [8] Honig, W. L., "Teaching and assessing programming fundamentals for non majors with visual programming", Proceedings of the 18th ACM conference on Innovation and technology in computer science education, Canterbury, England, UK, 2013.
- [9] Howland, K. & Good, J., "Learning to communicate computationally with Flip: A bi-modal programming language for game creation.", Computers & Education, 80:224-240 (2015).
- [10] McCracken, M., V. Almstrum, D. Diaz, M. Guzdial, D. Hagan, Y. Kolikant, C. Laxer, L. Thomas, I. Utting, & T. Wilusz., "A multi-national, multi-institutional study of assessment of programming skills of first-year CS students." In Working Group Reports of the 6th Annual Conference on Innovation and Technology in Computer Science Education, Canterbury, UK, ACM Press, 125-180 (2001).
- [11] Morelli, R., de Lanerolle, T., Lake, P., Limardo, B., Tamotsu, B., & Uche C., "Can Android App Inventor Bring Computational Thinking to K-12?", Unpublished paper. http://hfoss.org/uploads/docs/appinventor_manuscript.pdf, 2010.
- [12] Özding, F. ve Altun, A., "Bilişim Teknolojileri Öğretmeni Adaylarının Programlama Süreci ni Etkileyen Faktörler." İlköğretim Online, 13(4), 1531-1541, (2014).
- [13] Özmen, B. & Altun, A., "Undergraduate Students' Experiences In Programming: Difficulties And Obstacles." Turkish Online Journal of Qualitative Inquiry, 5(3), 9-27, (2014).
- [14] Powers, K. Ecott, S. & Hirshfield, L., "Through the Looking Glass: Teaching CS0 with Alice." Proceedings on the conference on Special Interest Group on Computer Science Education (SIGCSE) (2007).
- [15] Soloway, E., Bonar, J. & Ehrlich, K., "Cognitive strategies and looping constructs: An empirical study." Communications of the ACM, 26(11): 853-860 (1983).
- [16] Spertus, E., Chang, M.L., Gestwicki, P. & Wolber D., "Novel Approaches To Cs 0 With App Inventor For Android.", In Proceedings of the 41st ACM technical symposium on Computer science education (SIGCSE '10). ACM, New York, NY, USA, 2010.
- [17] Spohrer, J.C. & Soloway, E., "Putting it all together is hard for novice programmers." In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics,. Tucson, AZ, 1985.
- [18] Statita, "Number of apps available in leading app stores as of July 2014" <http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/> (2015).
- [19] Tigrek, S., & Obadat, M., "Teaching Smartphones programming using (Android Java): Pedagogy and Innovation", in Proc.ITHET. .1-7. (2012).
- [20] Wing, J. M., "Computational Thinking And Thinking About Computing", Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 366(1881): 3717-3725, (2008).
- [21] Wolber, D., "App Inventor and real-world motivation.", In Proceedings of the 42nd ACM technical symposium on Computer science education (SIGCSE '11), ACM, New York, NY, 2011.
- [22] Chadha, Karishma, & Turbak, Franklyn., "Improving App Inventor Usability Via Conversion Between Blocks And Text", Journal of Visual Languages & Computing, 25(6), 1042-1043, (2014).
- [23] Rubio, Miguel Angel, Romero-Zaliz, Rocio, Mañoso, Carolina, & de Madrid, Angel P., "Closing The Gender Gap In An Introductory Programming Course." Computers & Education, 82(0), 409-420, (2015).