

A gyakorlati programozás tanításának játékfejlesztésen alapuló, élménypedagógiai alapú módszerének bemutatása

Molnár György, Nyíró Péter

Budapesti Műszaki és Gazdaságtudományi Egyetem, Budapest

molnar.gy@eik.bme.hu, scorpeti@gmail.com

A programozás tanításának módszertani megközelítése

A számítástechnika, valamint az infokommunikációs technológiák nagymértékű fejlődésének köszönhető, hogy társadalmunk információközpontúvá vált (Siemens, 2005, 2006). Ennek hatására, ahogyan sok más diszciplína is, úgy a pedagógia tudományterülete is átalakulóban van (Benedek, 2007; Miskolczi, 2010). E változás a tanulás főszereplői számára is komoly kihívást jelent az oktatás legkülönbébb szinterein, így a felsőoktatásban is, ahol a diákoknak egyre nagyobb információmennyiséget kell elsajátítaniuk egyre rövidebb idő alatt (Molnár, 2011; Molnár & Benedek, 2013). Több kutatás bizonyította, hogy az emberi gondolkodás elsősorban vizuális alapú, ezt a képességét használva lényegesen nagyobb hatékonysággal tud elméleti ismereteket memorizálni és tanulni (Bleed, 2005; Nyíri, 2003; Visual Education, 2013). Ugyan sok helyen megjelenik már a „vizuális oktatás” gyakorlati alkalmazása, de jellemzően még ma is a hagyományos szöveg- és könyv alapú formák a leginkább elfogadottak és alkalmazottak a pedagógiában (Minden..., 2010).

Az informatikatanításnak – és ezen belül a gyakorlati programozás oktatásának – esetében ennek az ellentmondásnak egy szélsőséges változata tapasztalható. A számítógépek nagymértékű fejlődésének köszönhetően ma már nem okoz problémát a képek, hangok, animációk és videók megjelenítése bármilyen elektronikai berendezésen (Molnár, 2014). Ezzel szemben a programozás oktatása sok helyen még mindig a szöveges formát támogatja, kihasználatlanul hagyva az informatikában és a médiában rejlő bőséges vizuális potenciált. Mindez azért vált jelentős méretű problémává napjainkra, mert az információközpontú társadalomban az élet minden területén elengedhetetlen az informatikai tudás, sok helyen pedig a programozás alkalmazása is ugyanilyen. Éppen ezért szükséges jelentős figyelmet fordítani a programozás oktatására és a lehetőségekhez mérten hatékonyabb módszert alkalmazni a jelenleginél (Makó, 2013; Dobrovolná, 2015).

A célok meghatározása

Célunk a programozásoktatás egy olyan – Magyarországon újszerűnek mondható – tanítási módszerének bemutatása és gyakorlatba ültetése, amely teljes mértékben kihasználja a számítógépek adta vizuális lehetőségeket. A koncepció az elméletorientált oktatás helyett a gyakorlatra és a megszerzett tudás alkalmazhatóságára koncentrálni elsősorban. (Katona és Kurdi, 2008) A résztvevői motiváció növelésének érdekében az általunk kidolgozott oktatási módszer elsősorban a számítógépes játékfejlesztésre koncentrálni, középpontba helyezve a gamifikációt. (Haden, 2006; Kirkland és O’Riordan; 2008) A játékfejlesztés további előnye, hogy a programozáshoz szorosan kapcsolódó multimédiás és általános számítógépes ismeretek mind megjelennek a képzés során, ezért általában komplexebb figyelmet követel, mint egy funkcionális program tervezése. A szórakoztató ipar, mint célipar részére készülő komplex projektek az általános szoftverfejlesztéshez szükséges ismereteket és képességeket maradéktalanul tartalmazzák (Chen és Cheng, 2007; Rajaravivarma, 2005). Az alkalmazott oktatási módszer sikerességének vizsgálata érdekében rendszeresen indulnak kurzusok 2013 szeptembere óta.

Az új oktatásmódszertan kidolgozásának előzményei

A mai népszerű programnyelvek (például: Java, C, C++, C#, Objective-C, PHP, Python, Ruby) mindegyike szöveges formát használ. (Mikoluk, 2013) A szöveges forma előnye, hogy gyorsabban lehet vele dolgozni, mint a grafikus felületeket használó programnyelvekkel. Hátránya viszont, hogy egy adott programnyelv összes helyesírási formáját és elméleti felépítését meg kell tanulni, mielőtt azt széles körben alkalmazni lehetne. Egy kezdő programozó számára nehézséget okoz a szöveges programozás tanulása, mert az összes elméleti szabályt figyelembe kell vennie, mielőtt alkalmazni tudná a nyelvet. A szintaktika és a gondolkodásmód elsajátítása mellett egy-egy program speciális felépítésének tervezésére és az algoritmusok funkcionális alkalmazására már kevesebb figyelmet tud fordítani. Ezen komplex feladatok elvégzéséhez elengedhetetlen a programozásban alkalmazott logikai matematika ismerete és alkalmazása.

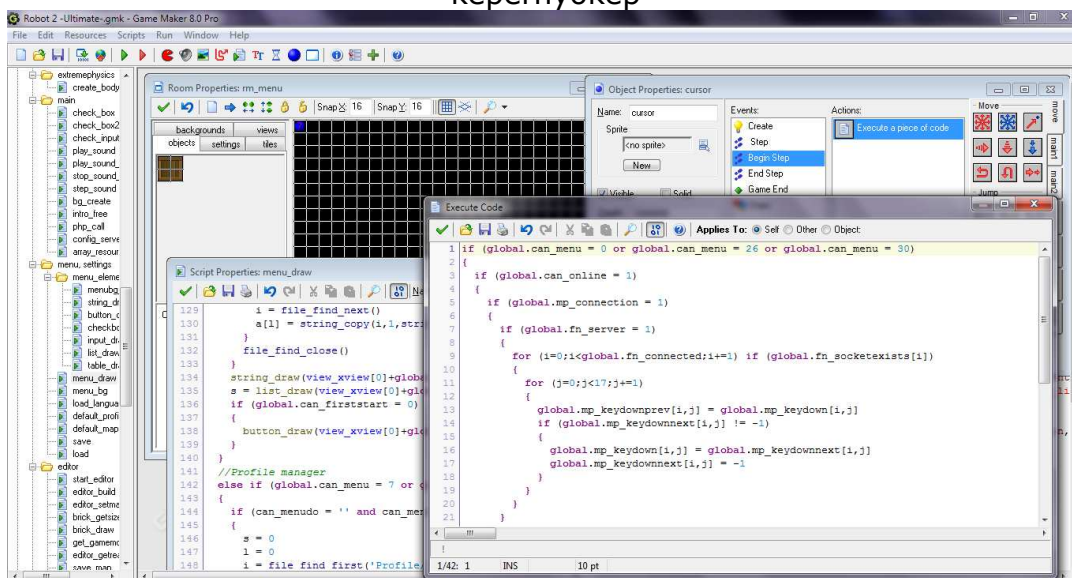
A grafikus programnyelvek (például: App Inventor, Scratch, LabVIEW) a szöveges nyelvekkel ellentétben egy kezdő programozó számára sokkal átláthatóbbak és könnyebben tanulhatóak. Ennek oka, hogy míg egy gyakorlott programozó a programsorok megírásakor sokszor fejben tartja a kódok kapcsolódási logikáját, a kezdőknek szükségük van annak vizuális megjelenítésére. A grafikus nyelvben írt program logikája a vizuális megjelenítés miatt könnyebben átlátható, továbbá szintaktikája is jóval egyszerűbb. Ebből következik, hogy egy grafikus programnyelv használatához kevesebb elméleti háttérre van szükség, a programtervezés mellett kevesebb figyelmet kell fordítani a helyesíráásra, könnyebb koncentrálni a program végső felépítésére. Megjegyzendő ugyanakkor,

hogy a grafikus programnyelvek felhasználásának keretei sokkal szűkebbek, alkalmazásuk pedig hosszú távon már nem olyan gyors, mint egy jól betanult szöveges programnyelvé.

Amíg tehát a kezdő programozóknak a grafikus programnyelvek tanulása a célravezetőbb, addig a tapasztaltabbak számára a szöveges forma az eredményesebb. A kettő között viszont sok programnyelv esetében nincs átmenet. A különböző fejlesztők által kiadott programozói szoftverek ilyen különbségei miatt a grafikus programnyelvből szövegesbe történő átállás általában nehézséget okoz, megnehezíti a különböző alkalmazások közötti tudás transzferálásának lehetőségét (Leutenegger & Edgington, 2004).

Az általunk kidolgozott tanítási módszer alapötlete számítógépes játékok vizsgálata alapján fogalmazódott meg. Főként a fiatal generációk esetében, a programozás játékon keresztül történő oktatása a számítógéphasználat adott komplexitása miatt kézenfekvő módszernek ígérkezett (Pap-Szigeti & Pásztor, 2008; Brain, 2009). Több programot is írtak már a célból, hogy valamilyen vizuális-játékos módszerrel próbálják a programozás logikáját megmutatni és a tanulóval begyakoroltatni (Anderson & McLoughlin, 2007; Lockhart, 2013; Pinola, 2013; Romano, 2013; Finley, 2013, Heussner, 2013; Kurzweil, 2013; Primerlabs, 2013). Hatékonyság szempontjából valóban jó eredményt mutat a legtöbb kezdeményezés, viszont a programokban található szűkített lehetőségek miatt ezek általában meg sem közelítik egy valódi programozási nyelv komplexitását (Teach..., 2013). Sajnos csupán néhány fejlesztői program alkalmas ténylegesen oktatási célra – ezekben a grafikus és a szöveges nyelv megfelelő módon, fokozatos áttérést biztosítva jelenik meg. A kurzusainkon alkalmazott Game Maker program e ritka szoftverek közé tartozik.

1. ábra. Játékfejlesztés a kurzusokon alkalmazott Game Maker 8.0 használatával. Elöl szöveges, hátul grafikus programozási felület látható, forrás: saját képernyőkép



Alkalmazott oktatásmódszertan

A felsőoktatásban megjelenő programozói kurzusok szinte teljes köre a hagyományosnak mondható, elméletorientált oktatásmódszertant részesítik előnyben az ismeretek átadása során. Eszerint a diákoknak előbb meg kell érteniük a programnyelv szintaktikáját, a struktúrák és algoritmusok működését, hiszen csak ezután lesznek képesek mindezt alkalmazni. Ez a módszertani megközelítés végül egy átfogó és részletes ismeretet ad az adott programnyelvről és annak használatáról. A hátránya, hogy az elméleti ismeretek elsajátítása a gyakorlati részt megelőzve nehezebb és emiatt lassabb folyamat. Mivel a megszerzett tudásnak a munka világában való alkalmazhatóságát főleg a gyakorlati feladatok határozzák meg, a rendelkezésre álló rövid idő alatt igazi kihívást jelent egy alkalmazható programozó részletes kitanítása a fenti ódon (Tóth & Johanyák, 2007).

A hagyományos oktatási módszertannal szemben áll a gyakorlatorientált programozás, amely egy erre kialakított felületen keresztül, vizuális hatásokkal szemlélteti az egyes funkciókat és lehetővé teszi, hogy alkalmazásokon keresztül sajátítsák el a diákok ugyanazt az elméleti tudást. Így rövidebb idő alatt alkalmazhatósági szempontból nagyságrenddel nagyobb anyagot lehet megtanítani, továbbá a gyakorlatok alapján a későbbi elméleti magyarázatok is könnyebben értelmezhetők.

Ennek megvalósítására választott Game Maker szoftver segítségével mind kezdő, mind haladó programozók számára biztosítani tudjuk a grafikus és szöveges programozói felületek közötti fokozatos átmenetet (Overmars, 2004). Begyakorolhatják a játékfejlesztéshez is szükséges alapvető programozói feladatokat és kihívásokat. A tanulás során nem csak elméleti tudásra tesznek szert, hanem szinte azonnal alkalmazni is fogják tudni. Az eszköz segítségével általános ismeretet szerezhetnek a C alapú objektumorientált programkódok elkészítéséről, az általános programtervezésről és a szükséges médiaismeretről egyaránt (Yoyo games, 2015).

2. ábra: A Game Maker 8.0 logója, forrás:
<http://logodatabase.net/game+maker+studio+logo>



A gyakorlatorientált oktatási módszer alkalmazása különböző típusú képzésekben

A kidolgozott oktatási módszert kurzusok keretei között lehet alkalmazni, bármely korosztálynál, a lifelong learning szemlélet jegyében. Egy kurzus terjedelme fél vagy egy év lehet az iskolarendszerhez, illetve iskolaszervezethez igazodva.

Felsőoktatásban vagy az iskolarendszeren kívüli oktatás alá tartozó felnőttképzésekben a kezdő kurzus fél év időtartamú, heti egyszeri másfél órás előadást jelent azok számára, akik korábban még sosem programoztak. A kurzus végére azok, akik emellett elvégezték a megfelelő mennyiségű otthoni munkát is, képesek lesznek önállóan saját grafikus szoftvert írni asztali számítógépre. A kezdő kurzusra épülő haladó kurzus szintén fél év időtartamú, heti egyszeri másfél órás előadást jelent, otthoni munkával kiegészítve. Itt szerzik meg a tanulók azt a tudást és képességet, amely segítségével saját programjukat a lehető legrövidebb idő alatt, optimális módon tudják elkészíteni (Karlovitc, 2015). Összességében tehát ezzel a módszertani megoldással egy év alatt meg lehet tanítani a korábban még soha nem programozó tanulókat a programozás alapjaira úgy, hogy azt alkalmazni is tudják a megszerzett tudásanyagot.

Az általános iskolák, szakközépiskolák és gimnáziumok oktatási rendjében való illesztés annyiban különbözik az itt ismertetett modelltől, hogy ott az otthoni, önállóan elvégzendő feladatokat lecserélik a tantermi gyakorlatok. Itt tanári ellenőrzés formájában azonnali visszajelzést kapnak a gyerekek munkájukkal kapcsolatban. A közös, felügyelt foglalkozások lassabb munkát eredményeznek az otthoni feladatmegoldáshoz képest, ennek következménye, hogy a kurzusok hossza fél év helyett egy tanév hosszúságúra növekszik, vagyis a kezdő és haladó kurzus együtt csak két tanév alatt teljesíthető.

Mindez persze csak akkor lehetséges, ha a tanulók motivációja a képzés és az önképzés során sem hagy alább. Ahhoz, hogy ez megvalósulhasson, a játékfejlesztés és a gamifikáció önmagában egy nagyon hatékony eszköz, de a tanár aktív figyelme is szükséges a megfelelő eredményhez. Az órákat kis létszámú, legfeljebb 15 fős csoportok számára ideális megtartani, hogy az oktatás személyesebb, közvetlenebb hangvétellé lehessen. A tanulók aktív figyelmére szükség van, hogy számítógépen keresztül képesek legyenek a tanár magyarázatát követni.

3. ábra. 2014 őszén, a BME-n tartott kurzus első órája, forrás: saját fénykép



Egy kurzus tantárgyi kerete célcsoporttól függően eltérő. Felsőoktatásban kurzusként illeszthető a kreditrendszerbe, közoktatásban szakkör formájában elérhető. A képzés kiegészítő tananyag tartalmát képezi továbbá az alapvető médiakezeléshez kapcsolható alapismeretek. A kurzusok követelményrendszerét otthoni munkák és projektek formájában érdemes meghatározni, mivel az alkalmazott programozást, mint képességet csak a megfelelő mennyiségű gyakorlás eredményeképpen lehetséges elérni.

A próbakurzusok elégedettségvizsgálatának bemutatása

A kutatási munka során több próbakurzust is indítottam konzulensem, Dr. Molnár György támogatásával; ezek célja az optimális oktatási modell minél nagyobb pontossággal történő kidolgozása volt. A próbakurzusok után elégedettségvizsgálatot végeztünk, levontuk a megfelelő következtetéseket, és kijelöltük a körvonalazódott további fejlesztési irányokat.

Az első próbakurzus egy internetes közösségi oldalon keresztül érte el a résztvevőit 2013 őszén, ami elsősorban felnőttek körében zajlott, ahol végül háromszorosra nőtt a túljelentkezés, köszönhetően a számítástechnika és a programozás mindennapokban betöltött szerepének. A kurzus igen vegyes területen dolgozó és tanuló résztvevőket is foglalkoztatott, ami következménye az előzetes tudás nagymértékű különbözősége és a jelentkezők előzetes elvárásainak jelentős szórása lett. Több BME, ELTE és SOTE hallgató is jelen volt az órákon, de érdeklődtek asztalos vagy műkőröm építő szakmában dolgozók is. Többek között ennek a változatosságnak köszönhető, hogy a kurzus előadásait módosítani kellett, a kezdő modell anyagának rövidített változatát tartalmazták, összesen 3 alkalomra redukálva. A három alkalom után az idő rövideje okán gyakorlatorientált számonkérés formájában került ellenőrzésre a hallgatók megszerzett tudása. Bár ennek sikeressége

erősen függött a hallgatók által befektetett otthoni munka mennyiségétől, elégedettség tekintetében egyértelmű pozitív változás mutatkozott. Az általuk kitöltött kérdőívek alapján rendkívül motiváltak lettek az oktatástechnológiát és a témát illetően.

A következő szemeszterben, 2014 tavaszán a próbakurzus a Budapesti Corvinus Egyetem, Kinizsi Kollégiumban került meghirdetésre egyetemisták és külsősök számára egyaránt, ahol a kezdő kurzus modelljének 8 alkalmas változatával ismerkedhettek meg az érdeklődők. Itt már házi feladatok formájában történt a számonkérés, amely kidolgozására két hónap állt rendelkezésre, majd utolsó előadáson prezentálni kellett a többi hallgató számára. A kurzus itt már nem csak elégedettségben mutatott rendkívül pozitív eredményeket, hanem szakmai tudás tekintetében is jelentős fejlődésen mentek keresztül a hallgatók. A kurzus egyértelmű sikerességét igazolja, hogy a hallgatók több mint negyede a házi feladatok hatására programozói pályát választott. Sok személy látott perspektívát a Game Maker program hosszú távú használatában és a további oktatási célú fejlesztésében. A visszajelzésekből az is kiderül, hogy akik végig járták az órákat, nagyon élvezték azt, hogy az idő során egyre mélyebben belelátnak egy programozó mindennapi problémáiba. Kivétel nélkül mindenki megemlítette, hogy az oktatásmódszertant alkalmazni kéne szélesebb körökben, mivel szerintük is hatékonyabban lehet megértetni ezzel a metodikával az algoritmikus gondolkozásmódot.

Az ezt követő két szemeszterben, 2014 őszén és 2015 tavaszán a Budapesti Műszaki és Gazdaságtudományi Egyetemen egy meglévő, szabadon választható tárgy tematikájába illesztve került feldolgozásra a téma, amely kurzusokon 6 alkalom erejéig került bemutatásra a kezdő modell rövidített változata. A számonkérés itt is házi feladat leadásának formájában történt, azonban a kurzus az egyetem formális kreditrendszerébe történt beillesztése miatt a célközönsége is jelentősen szűkült, így statisztikai adatokkal még nem lehet egyértelműen alátámasztani a hallgatók elégedettségét és a kurzus sikerességét. Ugyanakkor a legtöbb hallgató esetében a személyes visszajelzések alapján ez a kurzus is a korábbi mintát követi, több mérnökhallgató a leadott órák alapján kezdett el részletesebben foglalkozni a programozással. Többen is kifejtették személyes véleményüket azzal kapcsolatban, hogy a jelenleg működő programozás oktatás módszertana nem sikeres, mert bár nagy mennyiségű elméleti tudásra tehetnek szert a hallgatók, nem tudják azt rögzíteni gyakorlatokon keresztül. Az egyetemi kötelezettségeket ugyan teljesítik, de azok közel sem olyan kreatívak és motiváló hatásúak, mint a próbakurzusok által nyújtott játékfejlesztés. Sokan hozzátették, hogy a kurzusok sikerességéhez hozzájárult az is, hogy jómagam előadóként szakmailag felkészült voltam és pedagógiai szempontokból is helyálltam. A próbakurzusokon résztvevők elégedettsége tehát egyértelműen pozitív tendenciát mutatott.

Összegzés, kitekintés

A próbakurzusok pozitív eredménye egyértelműen látható. Ugyan a két éve zajló felmérés az ipari alkalmazhatóságot és az oktatási módszer hosszú távú hatását nem képes még kimutatni, már most látható a visszajelzések és a szorgalmi feladatokba fektetett energia alapján a motiváció ugrásszerű növekedése. Ennek hatására a hallgatók önszorgalomból, a szabadidejük jelentős részében is szívesen foglalkoznak szoftverfejlesztéssel, amely tudás az akár más területen végzett, számítógépes szoftverekkel kapcsolatos tevékenységüket segítheti.

Az eddigi eredmények alapján tehát levonható a következtetés, hogy érdemes ezzel a területtel tovább foglalkozni és szélesebb körben újabb felméréseket indítani. A kialakított új módszertani megközelítés fejlesztését erősen segítené, ha a későbbiekben induló kezdő kurzus az elméleti összefoglalóban vázolt javaslatoknak maradéktalanul megfelelné, illetve ha a haladó kurzus felmérése is megvalósulhatna.

Az új módszertan szélesebb körben történő alkalmazásához elengedhetetlen a megfelelő minőségű oktatói képzés is, hiszen a választott fejlesztői eszköz alkalmazásához szükség van a megfelelő szakmai háttérismeretekkel rendelkező pedagógusokra. A tanárok között egységesíteni kell a használatos oktatásmódszertant a tananyag fejlesztésének, illetve a kurzusok eltérő változatainak meghatározásának céljából, illetve a lehetőségekhez mérten rugalmasan alkalmazkodni a fellépő új igényekhez.

Hosszú távon, amennyiben a kutatás már kellő mennyiségű statisztikával tud szolgálni, érdemes összehasonlítani a hagyományos programozás oktatási módszerrel és a kiértékelés alapján konklúzióra jutni az oktatásmódszertant illetően. A jövőbeli cél, hogy a kutatómunka eredményeképpen a hallgatók elégedettségének javítása mellett javuljon a programozás oktatásának minősége és hatékonysága egyaránt.

Ezen célok megvalósítása érdekében várhatóan 2016 tavaszán a kifejlesztett oktatásmódszertan az egyetemi oktatási szintér mellett budapesti általános iskolákban és gimnáziumokban is meg fog jelenni tanítási órákon kívüli szakkörök formájában, amely kezdeményezés célja a digitális bennszülöttek informatikai képzésének pótlása, javítása (Edu&Joy, 2015). Emellett természetesen folytatódik az egyetemi oktatásban véghezvitt kutatás és a módszertan további fejlesztése és a statisztikai adatok gyűjtése. Hosszú távon kellő mennyiségű adattal rendelkezve már összehasonlítás végezhető az általunk alkalmazott és a hagyományos programozásoktatási módszer. A cél, hogy a munka eredményeképpen egyaránt javuljon a hallgatók elégedettségi szintje, valamint a programozás oktatásának minősége és hatékonysága.

Irodalomjegyzék

- Anderson, E. F., & McLoughlin, L. (2007). *Critters in the Classroom: A 3D Computer Game-Like Tool for Teaching Programming to Computer Animation Students*. ACM SIGGRAPH 2007 educators program, Article No. 7. New York, NY: ACM Publications.
- Benedek András (2007). Mobiltanulás és az egész életen át megszerezhető tudás. *Világosság*, (9), 21-28.
- Bleed, R. (2005). *Visual Literacy in Higher Education*. Maricopa Community Colleges, ELI Explorations, August 2005. Educause Learning Initiative.
- Brain, M. (2009). *Teaching your kids how to write computer programs*. <http://marshallbrain.com/kids-programming.htm> [2013.10.27.]
- Chen, W. K., & Cheng, Y. C. (2007). Teaching Object-Oriented Programming Laboratory With Computer Game Programming. *IEEE Transactions on Education*, 50 (3), 197-203.
- Dobrovolná, Alena (2015). Using Interactive Board and Communication in Teaching English at Lower-Secondary Stage of Elementary School. *Practice and Theory in Systems of Education*, 10 (1), 16-22.
- Edu&Joy (2015). <http://eduandjoy.hu/> [2015.08.01.]
- Finley, K. (2013). *World of Codecraft: 3-D Game Teaches Kids 'Big Ideas' of Programming*. <http://www.wired.com/wiredenterprise/2013/07/programming-game-engagement/> [2013.10.27.]
- Haden, P. (2006). The Incredible Rainbow Spitting Chicken: Teaching Traditional Programming Skills Through Games Programming. Eighth Australian Computing Education Conference (ACE2006). *Conferences in Research in Practice in Information Technology*, Vol. 52.
- Heussner, K. M. (2013). *Robots and rocket launchers: Kuato Studios uses gaming to teach kids the basics of coding*. Letöltés: <http://gigaom.com/2013/03/26/robots-and-rocket-launchers-kuato-studios-uses-gaming-to-teach-kids-the-basics-of-coding/> [2013.10.27.]
- Katona Krisztina, & Kurdi Zsombor Zsolt (2008). A Szoftvertervezés Gyakorlati Oktatása. In Pethő Attila, & Herdon Miklós (szerk.), *Informatika a felsőoktatásban, 2008 konferencia, Debrecen, 2008. augusztus 27-29*. 7 p. <http://www.agr.unideb.hu/if2008/kiadvany/papers/D11.pdf> [2016.02.03.]
- Kirkland, D., & O'Riordan, F. (2008). *Games as an Engaging Teaching and Learning Technique: Learning or playing?* Dublin: Griffith College.
- Kurzweil (2013). *A video game that teaches how to program in Java*. <http://www.kurzweilai.net/a-video-game-that-teaches-how-to-program-in-java> [2013.10.27.]
- Leutenegger, S., & Edgington, J. (2004). A Games First Approach to Teaching Introductory Programming. In *SIGCSE '07 Proceedings of the 38th SIGCSE technical symposium on Computer science education* (pp. 115-118). New York, NY: ACM Publications.
- Lockhart (2013). *Games That Teach Programming: A Brief Overview*. http://www.gamasutra.com/blogs/RobLockhart/20130905/199667/Games_That_Teach_Programming_A_Brief_Overview.php [2013.10.27.]
- Karlovitz János Tibor (2015). Az andragógiai kutatások néhány nyugat-európai irányzata. In Torgyik Judit (szerk.), *Százarcú pedagógia* (pp. 457-464). Komárno: International Research Institute.
- Makó Ferenc (2013). Esettanulmányos oktatás alkalmazása gyakorlatvezető mentortanárok képzésében. In Karlovitz János Tibor, & Torgyik Judit (szerk.), *Neveléstudományi és Szakmódszertani Konferencia. Vzdélavacia, výskumná a*

- metodická konferencia. Komárno, 2013. január 7-8. Program.* (p. 76.). Komárno, International Research Institute.
- Mikoluk, K. (2013). *Best Programming Languages to Learn in 2013: The Elite Eight.* <https://www.udemy.com/blog/best-programming-language/> [2013.10.27.]
- Minden, ami Közgáz (2010). *Oktatás + módszertan = oktatásmódszertan.* http://kozgaz.blog.hu/2010/06/17/oktatas_modszertan_oktatasmodszertan [2013.10.27.]
- Miskolczi Ildikó (2010). Tanulás vagy E-tanulás? – Oktatásmódszertan a XXI. században. *Szolnoki Tudományos Közlemények*, 14. http://tudomany.szolnok-ntesz.hu/kulonszamok/2010/cikkek/Miskolczi_Ildiko.pdf [2015.06.25.]
- Molnár Gy., & Benedek A. (2013). ICT Related Tasks and Challenges In The New Model of Technical Teacher Training. In John Terzakis, Constantin Paleologu, & Tibor Gyires (Eds.), *InfoWare 2013: ICCGI 2013: The Eighth International Multi-Conference on Computing in the Global Information Technology* (pp. 40-44.) Nice.
- Molnár György (2014). Pedagógiai megújulás tapasztalatai a szakmai tanárképzésben – új IKT alapú eszközök és koncepciók a tanárképzésben. In Ollé János (szerk.), *VI. Oktatás-Informatikai Konferencia Tanulmánykötet* (pp. 434-452). Budapest: ELTE PPK Neveléstudományi Intézet.
- Molnár György (2011). Új módszerek a pedagógiai gyakorlatban – az IKT alapú megoldások tükrében. *Szakképzési Szemle*, 27 (3), 170-177.
- Nyíri, K. (2003). From Texts to Pictures: The New Unity of Science. In Nyíri, K. (Ed.), *Mobile Learning: Essays on Philosophy, Psychology and Education* (pp. 45-67). Vienna: Passagen Verlag.
- Overmars, M. (2004). Teaching Computer Science through Game Design. *Computer*, (4), 81-83.
- Pap-Szigeti Róbert, & Pásztor Attila (2008). A LEGO Programozható Robotjaival Segített Programozásoktatás Beválasztóvizsgálata. In Pethő Attila, & Herdon Miklós (szerk.), *Informatika a felsőoktatásban* (p. 115). [Konferencia kiadvány.] Debrecen: Debreceni Egyetem Informatika Kar. <http://www.agr.unideb.hu/if2008/dok/absztrakt.pdf> [2015.06.25.]
- Pinola, M. (2013). *How and Why to Teach Your Kids to Code.* <http://lifehacker.com/how-and-why-to-teach-your-kids-to-code-510588878> [2013.10.27.]
- Primerlabs (2013). <http://primerlabs.com/> [2013.10.27.]
- Rajaravivarma, R. (2005). A games-based approach for teaching the introductory programming course. *Newsletter ACM SIGCSE Bulletin*, 37 (4), 98-102.
- Romano, B. (2013). *Dan Shapiro Builds a Board Game to Teach 3-Year-Olds Programming.* <http://www.xconomy.com/seattle/2013/09/03/dan-shapiro-builds-board-game-teach-3-year-olds-programming/> [2013.10.27.]
- Siemens, G. (2005). *Connectivism: A Learning Theory for the Digital Age.* http://www.itdl.org/Journal/Jan_05/article01.htm [2013.10.27.]
- Siemens, G. (2006). *Knowing Knowledge.* http://www.elearnspace.org/KnowingKnowledge_LowRes.pdf [2015.06.25.]
- Tóth György Ferenc és Johanyák Zsolt Csaba (2007). A szoftverfejlesztés oktatásának reformja a versenyszféra visszajelzései alapján. In V. *Pedagógiai Értékelési Konferencia, Szeged, 2007. április 12.*
- Visual Education (2013). <http://centerforvisualeducation.com/about-vised/new-blog-entry/> [2013.10.27.]
- Yoyo games (2015). <http://www.yoyogames.com/> [2015.09.05.]