

Algo-Ritmika: új dimenziók

Kátai Zoltán¹, Osztian Erika², Vekov Géza³, Osztian Pálma⁴

{¹katai_zoltan, ²osztian, ³vgeza}@ms.sapientia.ro, ⁴rozalia1997@yahoo.com
SAPIENTIA EMTE

Absztrakt. Az előadás keretében beszámolunk Algo-Ritmika projektünk több irányú kiterjesztéséről. A már jól ismert rendezési algoritmusokat illusztráló néptánc koreográfiáink kiegészítéseként, nyitottunk a keresési algoritmusok világa fele (lineáris, bináris, visszalépéses), illetve új táncstílusokat vezetünk be (flamenco, balett). Egy további újdonság, hogy egy újabb rendezési algoritmusnál a lineáris ábrázolásról kétdimenziósra váltottunk. Vizsgáltuk azt is, hogy az általunk fejlesztett interaktív e-learning környezetbe integrált algoritmus táncok, miként aknázhatók még hatékonyabban ki vegyes oktatási környezetben (blended learning).

Kulcsszavak: számítógépes gondolkodás, algoritmusok, művészet, tudomány, vegyes oktatási környezet

Bevezető

Az utóbbi években egyre nagyobb hangsúlyt kap a számítógépes gondolkodás fejlesztésének fontossága. Wing 2006-ban megjelent cikke különösen nagy lendületet adott ennek az irányzatnak. A kutatók egyre inkább azt hangsúlyozzák, hogy a három alapvető készség mellé, amelyet a társadalom minden tagjának birtokolnia kell, nevezetesen az írás, olvasás és számolás mellé, szükséges egy negyediket is felvenni: a számítógépes gondolkodást. [1] Wing megközelítésében a számítógépes gondolkodás egy olyan gondolkodásmód, amely révén a feladatokat úgy definiáljuk, és megoldásukat úgy közelítjük meg, hogy az eredményekhez egy információ feldolgozó egység révén tudjunk eljutni. Sőt, Wing árnyalja azt is, hogy mit jelent a számítógépes gondolkodás mindenki részére, és mit tudósok, mérnökök és más szakmabeliek részére. [2]

Elsőként az Egyesült Államokban próbálkoztak azzal, hogy a felsőoktatás különböző ágazataiba (nem csak az informatikus képzés keretében) számítógépes gondolkodáshoz kapcsolódó fogalmakat szöjjenek bele. Mivel ez még mindig csak egy szűk szegmensen jelentett, a figyelem a középiskolai, sőt a gimnáziumi szintű oktatásra is áttért, hogy a társadalom egésze célközönséggé válhasson. A [3] tanulmány szerzői külön felhívják a figyelmet azokra a tanulókra, akik továbbtanulásukat nem reál pályán képzelik el, illetve a lányokra, akik köztudottan alulképviselek az informatika szakterületen.

A CSTA [4] és ISTE [5] oktatás, illetve informatikaoktatás iránt elkötelezett szervezetek kilenc kulcsfogalmat társítanak a számítógépes gondolkodáshoz: adatgyűjtés, adatelemzés, adatábrázolás, részfeladatra bontás, absztrakció, algoritmus, automatizálás, párhuzamosítás, szimuláció. Az ISTE néhány készségeket is megfogalmazott a számítógépes gondolkodással kapcsolatban, köztük azt is, amikor valaki képes algoritmus bonyolultsági szempontokból összehasonlítani lehetséges megoldásokat. Nyilvánvaló, hogy e fogalmakhoz kapcsolódó kognitív műveletek fejleszthetők és gyakorolhatók bármely tantárgy oktatása keretében. Ezzel összhangban a [6] tanulmány azt vizsgálta, hogy már az elemi és gimnáziumi szintű oktatásban is miként tehető ez meg.

Bár, amint már említettük, a számítógépes gondolkodás fejlesztése nem föltétlenül feltételez kódolást, azért a megfelelő programozás-oktatás nyilván egy sajátos módja lehet e képesség

fejlesztésének. Többek között ez az oka annak, hogy egyre nagyobb hangsúlyt kap a felsőoktatás egyre több tudományterületén a programozás-oktatás. Sőt egyre többen hangsúlyozzák a programozás-oktatás fontosságát már középiskolai szinten, sőt léteznek kezdeményezések már elemi szinten is.

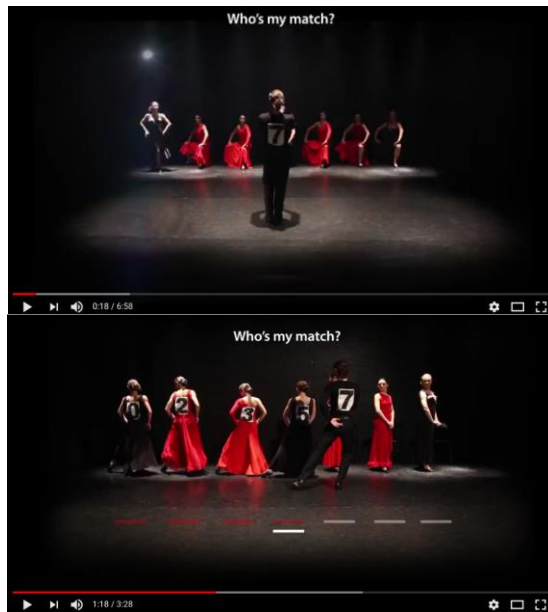
A mi Algo-Ritmika (AlgoRhythmics) projektünk középúton van a kódolás és az informatikától elvonatkoztatott számítógépes gondolkodás fejlesztés között: mindennapi élethez közelálló számítógépes algoritmusokat értetünk meg kódolás-mentesen, művészi elemekkel tarkított interaktív, e-learning környezetben.

Algo-Ritmika projekt

Az Algo-ritmika projektünk több, mint tíz évvel ezelőtt indult, és arra az alapgondolatra épült, hogy miként lehet több érzékszervet bevonni a programozás oktatásba. 2011-ben közzétettünk 6 videót, amelyek Erdélyben honos néptáncokkal illusztráltak rendezési algoritmusokat (www.youtube.com/user/AlgoRhythmics). A videókat egy online környezetbe ágyaztuk (<http://algo-rhythmics.ms.sapientia.ro/>), amely tánckoreográfiák mellé absztrakt animációkat társít, és lehetővé teszi az aktív tanulást, az algoritmusok interaktív levezénylése révén.

A bátorító nemzetközi visszhang, illetve a módszer hatékonyságát alátámasztó tanulmányok nyomán, úgy döntöttünk, hogy több vonalon is továbblépünk. A kiterjesztést igyekeztünk a már megalapozott elvek mentén megvalósítani. A dolgozat címében szereplő „új dimenziók” kifejezéssel az alábbiakra szerettünk volna célozni:

- 1) A projekt algoritmikai dimenzióját kiterjesztettük a keresési algoritmusok világára: lineáris, bináris, visszalépéses;
- 2) Az eddigi koreográfiáink lineáris ábrázolásban mutatták be, adott számsorozaton, a szóban forgó algoritmust, most viszont egy újabb rendezési stratégia kapcsán (kupac-rendezés), kétdimenziós ábrázolást alkalmazunk (az is újdonság, hogy táncospárok képviselik az egyes számokat);
- 3) A művészeti vonalon is tovább léptünk új táncstílusok felé: flamenco és balett (tervezünk hip-hop koreográfiákat is);
- 4) Egy kísérletben azt vizsgáltuk, hogy miként lehet hatékonyabban kiaknázni a megtervezett tanulási környezetben rejlő potenciált, amennyiben a tanár jól megválasztott kérdésekkel megvezeti a tanulók gondolkodását (blended learning).



1. **ábra:** Lineáris és bináris keresés flamenco táncal. (a) Kezdődik az első tánc. (b) Az első tánc nyomán.



2. **ábra:** Kupacrendezés kalotaszegi párossal, kétdimenziós ábrázolásban.



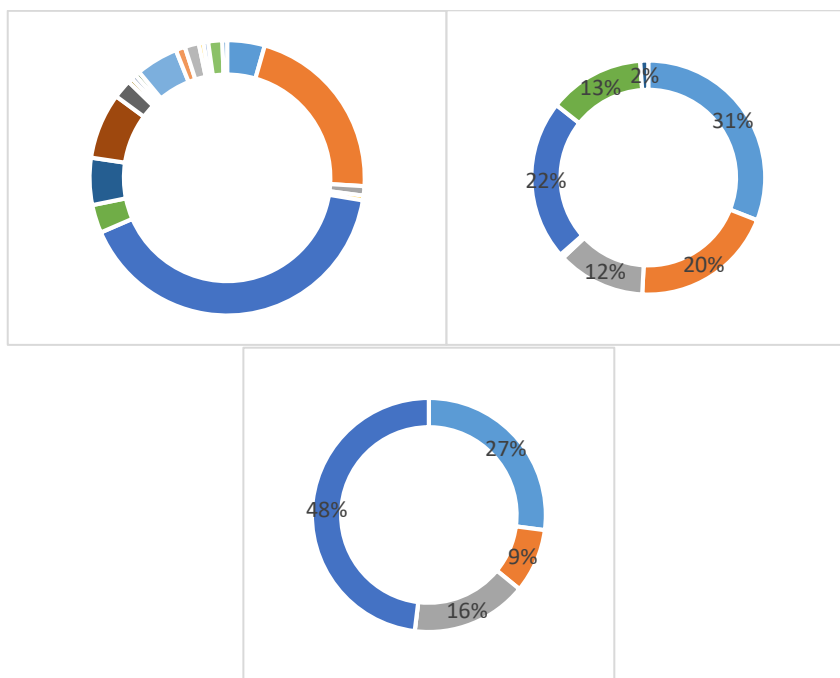
3. ábra: A 4-királynő feladatot megoldó visszalépéses keresési algoritmus balett koreográfiáját gyakorolják.

Kísérletek

Egy 5 lépéses vizsgálatot terveztünk meg. Azt vizsgáltuk, hogy a Sapientia EMTE, 2017-18-as tanévben elsőéves informatikus és mérnökképzős (számítástechnika, automatizálás, távközlés, mechatronika, gépészmérnöki) hallgatók mennyit tudtak meríteni megújult tanulási környezetükből. A kísérletre azt megelőzően került sor, hogy a diákok elkezdték volna a Programozás-I előadás hallgatását.

- (1. lépés) Az 1. mérésben (181 alany) arra összpontosítottunk, hogy mekkora potenciál van a tanulási környezetben (a) algoritmus megértetés, illetve (b) algoritmus bonyolultsági fogalmakra való rávezetés tekintetében, valamint hogy (c) milyen mértékben járulnak hozzá a környezet egyes elemei (tánc-koreográfiák, számítógépes animáció, interaktivitás) a nyújtott „algoritmika élményhez”;
- (2. lépés) A tanár, kiscsoportokban, megbeszélte a hallgatókkal az első kísérlet algoritmus bonyolultsági feladatait, ügyes kérdésekkel rávezetve őket a helyes válaszokra, és ezek miértjeire;
- (3. lépés) E 2. mérésnél arra fókuszáltunk, hogy a programozási előismeretekkel nem rendelkező hallgatók mennyit merítettek az előző lépések implementált vegyes tanulási környezetből, azaz képesek-e szignifikánsan jobban teljesíteni, algoritmus bonyolultsági feladatokon, további algoritmusokon?
- (4-5. lépések) A 3. és 4. mérések a tánc-koreográfiák és a számítógépes animáció relatív szerepét vizsgálták a létrehozott tanulási környezetben.

Az kísérletben résztvevő 181 alany rendkívüli sokszínűséget mutatott, ami a középiskolai előképzésüket illeti. Ezt tükrözi 4. ábra. E sokszínűség lehetőséget adott arra, hogy a „számítógépes gondolkodás mindenkinek” elv szemszögéből elemezzük ki az eredményeket.



4. ábra: A kísérleti alanyok (a) középiskolai szakirány szerinti bontásban; (b) egyetemi szakirány szerinti bontásban (világoskék: informatikus; többi szín: mérnökképzős); (c) programozási előképzés szerinti bontásban (világoskék: 0 év; barna: 1 év; szürke: 2 év; sötétkék: 4 év).

Az 1. mérésnél az alanyokat az alábbi tanmenetnek vetettük alá (a kérdéseket tömörítetten, kivonatolva közöljük):

- Megtekintették a lineáris keresés flamenco tánckoreográfiát (7 elemű számsorozatra; az oktató számítógépjéről kivetítve)
- Választak papíron az alábbi 7 kérdésre:
 - (K1.1.1-3) Hány összehasonlítást feltételez a lineáris keresés legjobb/legszerencsésebb esetben $7/31/N$ elemű számsorozat esetén?
 - (K1.1.4-6) Hány összehasonlítást feltételez a lineáris keresés legrosszabb/legszerencsétlenebb esetben $7/31/N$ elemű számsorozat esetén?
 - (K1.1.7) Mi számít a legrosszabb esetnek lineáris keresésére nézve?
- Megtekintették a bináris keresés flamenco tánckoreográfiát (7 elemű számsorozatra; az oktató számítógépjéről kivetítve)
- Választak papíron az alábbi 7 kérdésre:
 - (K1.2.1-3) Hány összehasonlítást feltételez a bináris keresés legjobb/legszerencsésebb esetben $7/31/N$ elemű számsorozat esetén?
 - (K1.2.4-6) Hány összehasonlítást feltételez a bináris keresés legrosszabb/legszerencsétlenebb esetben $7/31/N$ elemű számsorozat esetén?
 - (K1.2.7) Mi számít a legrosszabb esetnek bináris keresésére nézve?

- Megtekintették a buborékrendezés csángó néptánc-koreográfiát (10 elemű számsorozatra; az oktató számítógépéről kivetítve)
- Megtekintették a buborékrendezés számítógépes animációját (10 elemű számsorozatra; az oktató számítógépéről kivetítve)
- Válaszoltak papíron az alábbi 8 kérdésre:
 - (K1.3.1-3) Mi lenne az algoritmus első 3 művelete a generált (és kivetített) *látható* számsorozat esetén: Hasonlít/Cserél(a[...],a[...])? (Amint a hallgatók leírták a kurrens műveletet, a tanár elanimáltatta a kivetített képernyőn)
 - (K1.3.4-6) Mi lenne az algoritmus első 3 művelete a generált (és kivetített) *rejtett* számsorozat esetén: Hasonlít/Cserél(a[...],a[...])? (Amint a hallgatók leírták a kurrens műveletet, a tanár elanimáltatta a kivetített képernyőn)
 - (K1.3.7) Hány összehasonlítást feltételez annak kiderítése, hogy a 10 elemű számsorozat már eleve növekvő sorrendbe rendezett (legjobb eset)?
 - (K1.3.8) Hány cserét feltételez egy csökkenő sorrendbe rendezett (legrosszabb eset) 10 elemű számsorozat növekvő sorrendbe való átrendezése?
- (K1.4) Mikor éri meg a rendezés+bináris keresés stratégiát használni a lineáris keresés helyett? ((1) hosszú számsorokon; (2) sokszori keresés esetén; (3) nagy értékeket tartalmazó számsorokon; (4) egyéb)
- (K1.5.a,b,c) Mekkora fontossággal bírtak (1..7 skálán) az alábbi elemek az iménti „algoritmika élményben” ((a)tánc/(b)animáció/(c)kérdések)?

E mérést követően a tanár megbeszélte a hallgatókkal az algoritmus bonyolultsági kérdéseket. Rávezette őket arra (nagy hangsúlyt fektetve a miértekre), hogy mit jelentenek a „legjobb eset” és „legrosszabb eset” fogalmak, mi a szerepük algoritmus elemzési szempontból, illetve mik számítanak legjobb és legrosszabb eseteknek a keresési és rendezési algoritmusok szemszögéből.

A 2. mérésnél azt vizsgáltuk, hogy azok az alanyok, akik nulla évet tanultak programozást középiskolában, hogyan teljesítenek egy újabb algoritmuson. A kiválasztó rendezés tánc-koreográfiájának és animációjának megtekintése után felkértük a hallgatókat, hogy válaszolják meg a következő négy kérdést: hány összehasonlítást/cserét végez az algoritmus legjobb/legrosszabb esetben egy 10 elemű számsorozaton?

A továbbiakban az tánc-koreográfiák és az animáció relatív hozadékát próbáltuk lenyomozni. A következő méréseknél a hallgatók egy véletlen számsorozaton kellett megadják a bemutatott algoritmusok lépéssorozatát, kontrol csoportos felállításban. A 3. mérésnél a kísérleti csoportnak a beszűrő rendezést eltáncolva és elanimálva, a kontrolnak pedig kétszer elanimálva (más-más véletlen számsorokon) mutattuk be. A 4. mérésnél a shell-rendezés lett bemutatva nekik, az kísérleti csoportnak kétszer eltáncolva, a kontrolnak pedig négyszer elanimálva (így biztosítottuk, hogy a két csoport alanyai ugyanannyi időt töltsenek a „táncos”, illetve „animációs környezetben”).

Eredmények

1. mérés

Az első mérés nyomán kiszámítottuk, hogy minden egyes diák hány százalékosan teljesített az egyes kérdéscsoportok kapcsán, majd átlagot számoltunk először is az összes résztvevőre nézve (1. táblázat szürke háttérű sora). Ami az algoritmus bonyolultsági fogalmakra való ráérzést illeti

észrevehető, hogy a lineáris keresés esetében közel 90%-os volt a teljesítmény, a buborék rendezésnél közel 70%, a bináris keresésénél pedig közel 60% (mivel a K1.2.6 szinte senki sem válaszolt helyesen, ezért kizártuk elemzésünkéből). A legkisebb értéket, 58%-ot, a K1.4 kérdés kapcsán kaptuk, amikor is az alanyoknak mintegy szintetizálniuk kellett a bemutatott algoritmusokhoz kapcsolódó hatékonysági fogalmakat. A hallgatók az következő eseteket jelölték meg, mint legrosszabb/legszerencsétlenebb a keresésekre nézve: utoljára ellenőrzött elem a keresett (50%); hiányzik a keresett elem (41%); egyéb (9%).

A buborék rendezés algoritmus első három lépésének a rekonstruálásakor 88%-os teljesítményt nyújtottak a hallgatók. A generált véletlen számsorokon a helyes válaszok a következők voltak:

- Hasonlít(a[0],a[1]), Csere(a[0],a[1]), Hasonlít(a[1],a[2]) (a K1.3.1-3 kérdések kapcsán, látható számsoron);
- Hasonlít(a[0],a[1]); Hasonlít(a[1],a[2]); Csere (a[1],a[2]) (a K1.3.4-6 kérdések kapcsán, láthatatlan számsoron).

Amint számítottunk is rá, az alanyok enyhén jobban teljesítettek a láthatatlan számsoron [7]. Szignifikáns eltérés csak a K1.3.1 és K1.3.4 kérdések kapcsán volt. Mivel a látható számsor esetében a hallgatók látták, hogy az első két számot ki kell cserélni, a 33%-uk átugorta az összehasonlítási műveletet. A K1.3.4 kérdésre több mint 90%-ka a hallgatóknak jól válaszolt.

Összegzésként kijelenthetjük, hogy ami a teljesítményt illeti, az alanyok viszonylag jól teljesítettek az algoritmus megértés tekintetében (K1.3.1-6), viszont gondjaik voltak algoritmus bonyolultsági fogalmakra való ráérzésnél, főleg kevésbé kézenfekvő esetekben: K1.2.4-6 (59%,35%,6%); K1.3.8 (62%); K1.4 (58%). Ezért úgy következtettünk, hogy főleg algoritmus bonyolultsági szempontok felszínre hozása tekintetében lenne indokolt a tanári jelenlét, aki további, rávezető kérdésekkel serkentheti a hallgatók megfelelő irányokba való gondolkodását. Ezt tartottuk szem előtt a vizsgálat következő lépéseinek megtervezésénél. Amint említettünk már, a 2. mérésnél azt vizsgáltuk, hogy miután a tanár megbeszélte a K1.1.1-7, K1.2.1-7, K1.3.7-8 és K1.4 kérdéseket a hallgatókkal, miként teljesítenek majd egy újabb algoritmus kapcsán.

Ami az egyes elemeknek a nyújtott algoritmika élményhez való hozzájárulását illeti, a megkérdezettek az alábbi sorrendet állították fel (K1.5.a-c oszlopok): tánckoreográfiák < interaktivitás-kérdések-révén < számítógépes-absztrakt-animáció (szignifikáns eltérésekkel). Elbeszélgetve a hallgatókkal arra éreztünk rá, hogy az animáció magas pontszámához főleg az járult hozzá, hogy ekkor került sor az algoritmus (buborék rendezés) tényleges megértésére. Elmondásuk szerint a tánckoreográfiáknak az érdeklődés felkeltésében volt nagyobb szerepe, illetve az algoritmusra való ráhangolásban. A tánckoreográfiáknak köszönhetően az animáció már felkészülten találta őket. Mindez arra motivált bennünket, hogy további mérésekkel vizsgáljuk a tánckoreográfiák és az animáció viszonylagos szerepét a szóban forgó tanulási környezetben.

	Teljesítmény					Értékelés		
	K1.1. 1-6	K1.2. 1-5	K1.3. 1-6	K1.3. 7-8	K1.4	K1.5.a	K1.5.b	K1.5.c
összes	88%	59%	88%	68%	58%	3.97	5.94	5.29
0 év	79%	34%	81%	59%	51%	3.61	6.02	5.42
1-2 év	86%	53%	88%	49%	52%	4.31	6.13	5.38
4 év	94%	76%	92%	83%	65%	3.97	5.80	5.17

1. táblázat: Az első mérés eredményei az összes alanyra vonatkoztatva, illetve programozási előismeretek (0/1-2/4 év középiskolai programozás oktatás) szerinti bontásban. Az oszlopok kérdéskategóriák szerinti átlagokat tartalmaznak.

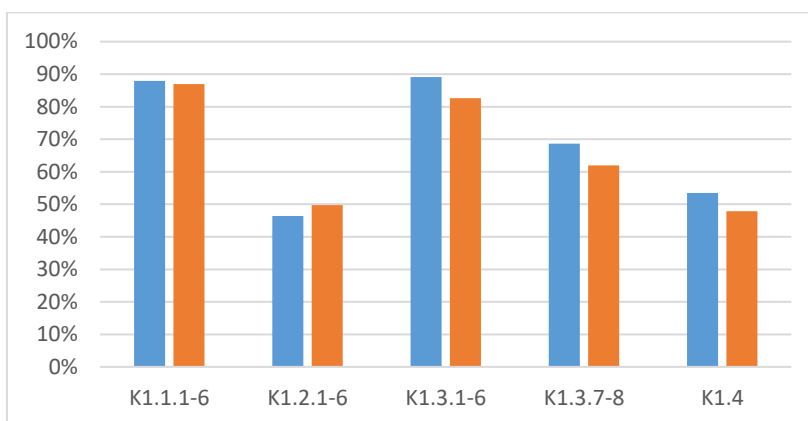
Programozási előismeretek szerinti bontásban

Vizsgáltuk azt is, hogy milyen volt a hallgatók teljesítménye/értékelése programozási előismeretek szerinti bontásban (1. táblázat utolsó három sora; a félkövér értékek szignifikánsan nagyobb értékeket jelölnek az előző sorhoz viszonyítva). Ami a K1.2-4 kérdéseket illeti, akik informatika osztályokból jöttek (középiskolában 4 évet tanultak programozást) következetesen jobban teljesítettek kollegáiknál. Nem ilyen egyértelmű az elhatárolódás a 0 és 1-2 év középiskolai programozás oktatás esetében. Érdekes, hogy csak a bináris keresés kapcsán teljesítettek szignifikánsan túl 1-2 évet tanultak a 0 évet tanultakat. Meglepő módon a K1.3.7-8 kérdéseknél a 0 évet tanultak túl teljesítettek a 1-2 évet tanultakat. Az is meglepő, hogy a K1.3.1-6 és K1.4 kérdéseknél nem voltak szignifikáns elhatárolódások.

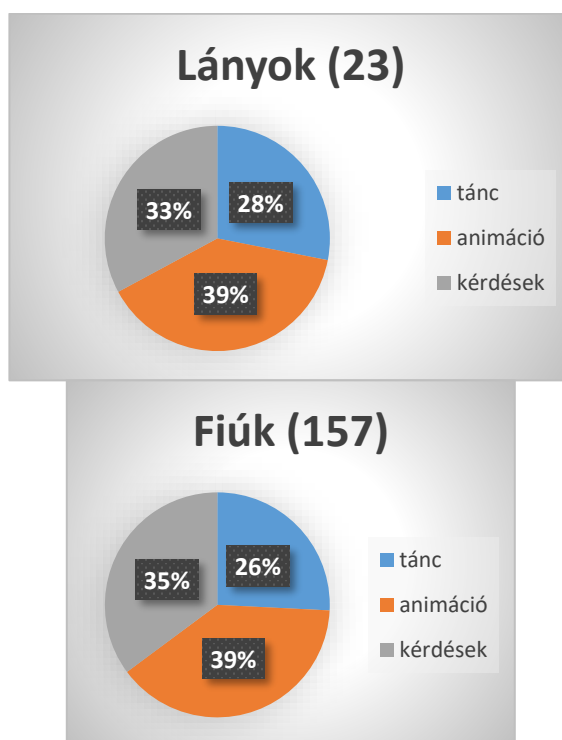
Ami a tánc/animáció/kérdések elemek értékelését illeti, az egyetlen szignifikáns eltérés a tánc tekintetében volt az 1-2 évet tanultak a 0 évet tanultak között. E mögött az állhat, hogy akik tanultak valamennyi programozást, de azzal maradtak, hogy nem igazán értették meg annak idején az algoritmusokat, inkább tudták értékelni a tánc-koreográfiák hozadékát a jelen tanulási környezetben. Akinek minden új volt (0 év programozási előzetes), azoknak nem volt viszonyítási alapjuk, akik 4 év programozással voltak a hátuk mögött, azok úgy érezhették, hogy nem igazán volt szükségük tánc-koreográfiákra a megértéshez.

Nemek szerinti bontásban

Amikor nemek szerinti bontásban (157 fiú, 23 lány, 1 nem adta meg a nemét) vizsgáltuk a hallgatók válaszait, nem talált szignifikáns különbségeket köztük semelyik kérdés-kategória tekintetében. Az 5. ábra a fiúk/lányok teljesítményét mutatja be, a 6. pedig tánc/animáció/kérdések elemek kapcsán adott értékelésüket. Érdekes, hogy mindkét csoport pontosan 69%-kosnak (az 1(0%)..7(100%) skálán) értékelte a kísérlet nyújtotta „algoritmus-élményt”.



5. ábra: Teljesítmények nemek szerinti bontásban (kék: fiúk, barna: lányok)



6. ábra: A K1.5.a-c kérdések kapcsán való értékelések nemek szerinti bontásban.

2. mérés

A 2. mérésben 46 olyan hallgató vett részt, akik nem tanultak programozást a középiskolában. A 2. táblázat közli ezek eredményeit az 1. mérés buborék rendezéshez és a 2. mérés kiválasztó rendezéshez kapcsolódó algoritmus bonyolultsági kérdésein.

	Legjobb eset (hasonlítások)	Legjobb eset (cserék)	Legrosszabb eset (hasonlítások)	Legrosszabb eset (cserék)
1. mérés	78%	-	-	40
2. mérés	51%	95%	71%	57%

2. táblázat: A nulla programozási előismeretekkel rendelkező alanyok teljesítményei az első két mérés kapcsán.

Az 1. mérésnél legjobb esetben csak az hasonlítások számára, legrosszabb esetben pedig csak a cserék számára kérdeztünk rá. Legrosszabb esetben, a cserék számát illetően, a kiválasztó rendezésen a hallgatók szignifikánsan jobban teljesítettek, mint a buborék rendezés esetében. Ez azt jelenti, hogy érzékelték, hogy bár más a stratégia ugyanannyi csere szükségeltetik. Érdekes, hogy szignifikánsan többen érzékelték jól, hogy hány hasonlításra van szükség (71%), mint akik helyesen adták meg a cserék számát (57%). Ez azt jelenti, hogy egy jelentős számú hallgatóban nem tisztult le, hogy a hasonlítások és cserék száma megegyezik legrosszabb esetben.

A kiválasztó rendezés sajátossága, hogy legjobb esetben is ugyanannyi hasonlítás szükségeltetik, mint legrosszabb esetben. Ez egy jelentősen elvontabb jelenség, minthogy a buborék rendezés (n-1) hasonlítás után felismeri, hogy a számsor eredetileg rendezett. Nem meglepő, hogy ebben a kategóriában a 2. mérésben alulteljesítettek a hallgatók. Sőt figyelemre méltó, hogy több mint felük (51%) jól adta meg a hasonlítások számát legjobb esetben. Szinte minden hallgató (95%) tisztában volt azzal, hogy legjobb esetben nem kerül sor cserére.

3. mérés

A 3. méréskor, egy újabb algoritmuson (beszűrő rendezés), azt vizsgáltuk, hogy mi a viszonylagos hozzáadéka a tánckoreográfiáknak versus számítógépes animáció. Azt figyeltük meg, hogy a kísérleti (56 alany) és kontrol (46 alany) csoportok szinte ugyanúgy teljesítettek. A tánc+animáció környezetben 72%-os, az animáció+animáció környezetben pedig 71%-os volt a teljesítés. Érdekes, hogy akik nem tanultak előzőleg programozást azok jobban teljesítettek a kontrol környezetben, akik viszont tanultak azok a kísérleti környezetben voltak eredményesek (a különbségek viszont nem voltak szignifikánsak).

4. mérés

A hallgatókkal való elbeszélgetés feltárta, hogy a tánc+animáció párosítás egy hiányossága, hogy mire megszoknák, változik a környezet. Ezért az utolsó mérésnél, a shell rendezés kapcsán, kétszer vetítettük le a tánckoreográfiát, és négyszer játszódtuk le az animációt, más-más számsorokon. Meglepetésünkre, a kísérleti csoport elsöprően túlteljesítette a kontrolt: 69% vs. 30% (igaz csak kis mintán: 21+23 alany). Mindez annak ellenére, hogy kontrol csoport összetétele programozási előismeret szerint (0 év: 65%; 1-2 év: 0%; 4 év: 35%) volt, míg a kísérleti (0 év: 76%; 1-2 év: 19%; 4 év: 5%).

Következtetések

A jelen tanulmány, a közölt kísérleti eredmények révén alátámasztja, hogy a bemutatott tanulási környezetben van potenciál (a tárgyalt algoritmusok megértésén túl) olyan elvont, számítógépes gondolkodáshoz kapcsolódó fogalmak közvetítéséhez is, mint az algoritmusok bonyolultsága. E tekintetében kritikusan fontos lehet a tanári jelenlét, aki ügyes kérdésekkel serkentheti a környezet ez irányú kiaknázását.

A vizsgálat aláhúzta a gondosan elkészített számítógépes animációk és az interaktivitás szerepének (legalább rávezető kérdések által) fontosságát a számítógépes algoritmusok oktatásánál. Felszínre jött továbbá az is, hogy bár minden újabb ábrázolás (tánc-koreográfia, absztrakt animáció) pluszokat hozhat a megértés szempontjából, hátráltató is lehet: a változás figyelemeltereléssel is jár. E jelenség jobb megértése további kutatást feltételez.

Az eredmények általánosítása tekintetében óvatosságra sarkal az, hogy egyes mérések viszonylag kis mintán születtek, illetve fenn áll az algoritmus függőség lehetősége is. Ez alatt azt értjük, hogy egyes algoritmusok esetében kifejezőbb lehet egy találó animáció, másoknál pedig egy jól átgondolt tánc-koreográfia. A tánc+animáció páros akkor tud igazán nyerő lenni, ha ki tudjuk küszöbölni az ábrázolás váltásból adódó esetleges mínuszokat. Meggyőződésünk, hogy a tánc-koreográfiák szerepe e környezetbe túlmutat a motiváció felkeltésén.

Irodalom

1. WING, J. M. 2006. Computational thinking. *Communications of the ACM*, 49(3), 33-35.
2. WING, J. M. 2011. *Computational thinking – what and why*. <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>
3. SETTLE, A., FRANKE, B., HANSEN, R., SPALTRO, F., JURISSON, C., RENNERT-MAY, C., & WILDEMAN, B. 2012. July. Infusing computational thinking into the middle-and high-school curriculum. In *Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education* (pp. 22-27). ACM.
4. ISTE. <http://www.iste.org/>, 2017.
5. CSTA. <http://csta.acm.org/>, 2017.
6. MANNILA, L., DAGIENE, V., DEMO, B., GRGURINA, N., MIROLO, C., ROLANDSSON, L., & SETTLE, A. 2014. Computational thinking in K-9 education. In *Proceedings of the working group reports of the 2014 on innovation & technology in computer science education conference* (pp. 1-29). ACM.
7. KATAI, Z. 2014. Selective hiding for improved algorithmic visualization. In *Proceedings of the 2014 conference on Innovation & technology in computer science education* (pp. 33-38). ACM.