



1949

**Interaktív, többérzékszerves elektronikus oktatási környezet a
számítógépes gondolkodás fejlesztéséhez**

Egyetemi doktori (PhD) értekezés

a szerző neve: Osztián Erika

témavezető neve: Dr. Fazekas Gábor

DEBRECENI EGYETEM

Természettudományi és Informatikai Doktori Tanács
Matematika-és Számítástudományok Doktori Iskola
Debrecen, 2021



1949

**Interaktív, többérzékszerves elektronikus oktatási környezet a
számítógépes gondolkodás fejlesztéséhez**

Egyetemi doktori (PhD) értekezés

a szerző neve: Osztián Erika

témavezető neve: Dr. Fazekas Gábor

DEBRECENI EGYETEM

Természettudományi és Informatikai Doktori Tanács
Matematika-és Számítástudományok Doktori Iskola
Debrecen, 2021

Ezen értekezést a Debreceni Egyetem Természettudományi és Informatikai Doktori Tanács Matematika- és Számítástudományok Doktori Iskola Szakmódszertan programja keretében készítettem a Debreceni Egyetem természettudományi/műszaki doktori (PhD) fokozatának elnyerése céljából. Nyilatkozom arról, hogy a tézisekben leírt eredmények nem képezik más PhD disszertáció részét.
Debrecen, 2021. 01. 29.

Osztian Erika

Tanúsítom, hogy Osztian Erika doktorjelölt 2016- 2020 között a fent megnevezett Doktori Iskola Szakmódszertan programjának keretében irányításommal végezte munkáját. Az értekezésben foglalt eredményekhez a jelölt önálló alkotó tevékenységével meghatározóan hozzájárult. Nyilatkozom továbbá arról, hogy a tézisekben leírt eredmények nem képezik más PhD disszertáció részét.
Az értekezés elfogadását javasolom.

Debrecen, 2021. 01. 29.

Dr. Fazekas Gabor

**Interaktív, többérzékszerves elektronikus oktatási környezet a számítógépes
gondolkodás fejlesztéséhez**

Értekezés a doktori (Ph.D.) fokozat megszerzése érdekében
a matematika- és számítástudományok tudományágban

Írta: Osztián Erika okleveles informatikatanár

Készült a Debreceni Matematika- és Számítástudományok doktori iskolája
(Szakmódszertan programja) keretében

Témavezető: Dr. Fazekas Gábor

Az értekezés bírálói:

Dr.
Dr.
Dr.

A bírálóbizottság:

elnök: Dr.
tagok: Dr.
Dr.
Dr.
Dr.

Az értekezés védésének időpontja: 2021... .

Tartalomjegyzék

1. BEVEZETŐ.....	8
1.1. Számítógépes gondolkodás a XXI. században	8
1.2. Célkitűzések.....	10
1.3. TÉZISEK	12
1.4. Az értekezés áttekintése.....	14
2. Az AlgoRhythmics tanulási környezet	16
2.1. „Látni”, „hallani” és „tapintani” az algoritmusokat	17
2.2. Algo-ritmika (eltáncolt algoritmusok).....	18
2.3. Egy webes alkalmazás az algoritmusok oktatásához	20
2.4. Algoritmika mindenkinek.....	21
2.5. Az AlgoRhythmics projekt új lendületet kap	22
3. Sematikus versus emberi mozgást alkalmazó realisztikus algoritmusvizualizáció	28
3.1. Szakirodalmi áttekintés.....	28
3.2. Kutatási kérdések.....	33
3.3. KÍSÉRLET	34
3.4. Eredmények és kiértékelés	36
3.5. Következtetések.....	43
4. Sematikus és realisztikus algoritmusvizualizációk ismételt megtekintése, illetve ezek kombinálása	45

4.1. Szakirodalmi áttekintés.....	45
4.2. Kutatási kérdések.....	48
4.3. KÍSÉRLET	49
4.4. Eredmények és kiértékelés	50
4.5. Következtetések.....	54

5. Az interaktivitás szerepe online algoritmusvizualizációs környezetekben..... 56

5.1. Szakirodalmi áttekintés.....	56
5.2. Kutatási kérdések.....	59
5.3. KÍSÉRLET	59
5.4. Eredmények és kiértékelés	60
5.5. Következtetések.....	65

6. Tanári kérdéssorok algoritmusvizualizációs környezetekben..... 67

6.1. Szakirodalmi áttekintés.....	67
6.2. Kutatási kérdések.....	69
6.3. KÍSÉRLET	69
6.4. Eredmények és kiértékelés	73
6.5. Következtetések.....	77

7. Számítógépes gondolkodás vizsgálata különböző korosztályú tanulóknál 79

7.1. Szakirodalmi áttekintés.....	79
7.2. Kutatási kérdések.....	82

7.3. KÍSÉRLET	82
7.4. Eredmények és kiértékelés	84
7.5. Következtetések.....	95
8. Limitációk	97
9. Összegzés	98
9.1. Új fejlesztések.....	98
9.2. Új kutatások.....	99
10. Summary	104
10.1. New results:	104
11. Következtetések	109
12. Köszönetnyilvánítás	110
13. Mellékletek.....	111
13.1. A kutatások menete	111
14. Függelék	129
14.1. A függelék: Számítógépes gondolkodás teszt	129
14.2. B függelék: A buborék- és kiválasztó rendezések pszeudókódja.....	132
15. Ábrajegyzék	133
16. Táblázatjegyzék.....	135
17. Irodalomjegyzék.....	136
18. Forrásjegyzék	141
19. Publikációs jegyzék	142

1. BEVEZETŐ

1.1. Számítógépes gondolkodás a XXI. században

Már a XX. század második fele is arról szólt, hogy a számítógépek egyre nagyobb teret hódítottak. Mostanra viszont ez a folyamat olyan szintre jutott, hogy a számítógépes gondolkodás a társadalom minden tagja számára alapkészség kellene legyen. Wing 2006-os nagy befolyású cikkében így fogalmaz: „A számítógépes gondolkodás, mint új műveltség a XXI. század meghatározó oktatási alapelvénvé, és tanulási képességévé válik” (Wing, 2006). Beindult egyfajta „Számítógépes gondolkodás mindenkinek” mozgalom, amelynek egyik közvetlen hozadéka, hogy az oktatás minden szintjén (elemi, gimnázium, középiskola, felsőoktatás) törekvések születtek a készség fejlesztésére.

Annak ellenére, hogy ennyire előtérbe került a számítógépes gondolkodás, még sincs egy letisztult definíciója. Wing (2006) úgy definiálja, hogy „A számítógépes gondolkodás olyan kognitív folyamat, amely magába foglalja a problémának, valamint lehetséges megoldásainak a megfogalmazását és a reprezentálását oly módon, hogy az hatékonyan végrehajtható legyen egy információfeldolgozó ágens által”. Aho (2012) újra értelmezi a fogalmat, és a következő meghatározást adja: „Computational thinking refers to the thought processes involved in expressing solutions as computational steps or algorithms that can be carried out by a computer.” Denning (2009, 2017) hangsúlyozza, hogy mennyire fontos az Aho definíció középpontjában álló számítógépes modell szerepe. Ő abból indul ki, hogy a számítógépes gondolkodás az algoritmikus gondolkodás fogalmából nőtt ki, és ebből kifolyólag, elsősorban, a számítógépes algoritmusoknak (nem bármely algoritmusnak) jut fontos szerep a számítógépes gondolkodás fejlesztésében. Felhívja a figyelmet

arra, hogy azok az algoritmusok, amelyekben szerepet kap az emberi döntéshozatal, nem tekinthetők számítógépes modelleknek.

Hogyan lehet összeegyeztetni e két nézőpontot, azt, hogy a számítógépes gondolkodás mindenkinek szükséges, és azt, hogy a számítógépes algoritmusoknak fontos szerepük van e készség fejlesztésében? Vajon a közös nevező az lenne, hogy mindenki tanuljon meg programozni? Igen, de nem feltétlenül a hagyományos értelemben vett programozást (bár ebbe az irányba haladunk) (Soloway, 1993; Ben-Ari, 2011). Egy lehetséges középút lehet, hogy a számítógépes algoritmusokat olyan kontextusban mutatjuk be, amely bárki számára vonzó tud lenni. Például az AlgoRhythmics kutatócsoportunk, amely a Sapientia Erdélyi Magyar Tudományegyetem keretében működik, a megoldást az informatika és a tánc párosításában látja. Művészegyüttesekkel karöltve, tánckoreográfiákat dolgoztunk ki, amelyek alapvető számítógépes algoritmusokat illusztrálnak.

Az AlgoRhythmics tanulási környezet, mint webalkalmazás, jelenleg hét rendezési és három keresési stratégiához tartalmaz dinamikus vizualizációkat: algoritmustáncokat és számítógépes animációkat. Amíg a tánckoreográfiák hozzájárulnak a többérzékszerves tanulási élményhez, az absztrakt ábrázolások segítenek az algoritmusok lényeges elemeire összpontosítani. Egy másik sajátossága a környezetnek, hogy különböző interaktivitási szinteket tesz lehetővé a felhasználók részére. Kijelenthető tehát, hogy az AlgoRhythmics platform, valóban egy interaktív, többérzékszerves elektronikus oktatási környezet a számítógépes gondolkodás fejlesztéséhez. A következőben felvezem a kutatási célokat, melyeket a dolgozat címében tömörített kutatási témát szem előtt tartva fogalmaztam meg.

1.2. Célkitűzések

Amíg az absztrakt animációk a sematikus ábrázolások kategóriájába sorolhatók, addig a tánckoreográfiákkal való ábrázolás inkább egyfajta realiztikus vizualizációnak tekinthető. Korábbi tanulmányok úgy következtetnek, hogy a sematikus ábrázolások általában hatékonyabbak, mert segítenek a vizsgálat szempontjából fontos elemekre figyelni. Ezzel szemben a realiztikus megjelenítések sok figyelemelterelő elemet tartalmazhatnak. A realiztikus ábrázolások erősségei között a motivációs szempontokat szokták leggyakrabban említeni (Carney & Levin, 2002), illetve azt, hogy segítik a tanultakra való visszaemlékezést. Tény, hogy a tánckoreográfiákat számos, potenciálisan figyelemelterelő elem díszíti: megkapó zene, élénk öltözet, magával ragadó tánc stb. Persze ezek multiszenzoriális elemekként is felfoghatók, amelyek pluszokat rejthetnek magukban a tanulási folyamatra nézve. A tánckoreográfiák egy másik erőssége lehet, hogy emberi mozgást alkalmaznak. Friss kutatások kiemelik, hogy az úgynevezett „emberi mozgás effektus” jelentősen hozzájárulhat a dinamikus vizualizációk hatékonyságához (Castro-Alonso et al., 2014). A jelen dolgozatban bemutatott *első* kutatás a „Sematikus versus emberi mozgást alkalmazó realiztikus algoritmus vizualizáció” témakört járja körül.

Érdekes, hogy a szakirodalom (Byrne & Lyons, 2001) vegyes eredményekről számol be, ami a statikus és dinamikus vizualizációk hatékonyságát illeti. A leggyakrabban felhozott ok, amiért az animációk nem következetesen hatékonyabbak, mint a statikus ábrázolások, az a dinamikus ábrázolások tranziens jellegéhez kapcsolódik. Egy kézenfekvő válasz e jelenségre, az ismételt megtekintés lehet. Az AlgoRhythms környezet viszont, az egyszerű megismétlés mellett, azt is lehetővé teszi, hogy a vizsgált algoritmust más-más ábrázolás-

ban tekintsék meg a diákok, ismételten. Ezért a *második* kérdés, amit megvizsgálunk az, hogy mi az egymáshoz viszonyított értékük különböző vizualizáció kombinációknak (sematikus követi a realisztikus; realisztikus követi a sematikus; csak sematikus, kétszer megjelenítve; csak realisztikus, kétszer megjelenítve).

A modern oktatás egyik alapeleme az aktív tanulás elve (Korhonen & Malmi, 2000). Ezen alapelve épülnek az interaktív oktatási környezetek. Az algoritmus vizualizációs eszközök általában kétféleképpen aktiválják a tanulót.

Egyik módszer, hogy a vizualizáció kulcsmomentumaiban megáll az animáció/videó, és a tanuló meg kell, hogy mondja a következő lépést, illetve meg kell, hogy válaszoljon bizonyos kérdéseket (nevezzük ezt „fél interaktivitásnak”). Egy másik lehetőség, amikor a felhasználó fel van kérve, hogy vezényelje le a teljes algoritmusvizualizációt („teljes interaktivitás”). Előző kutatások e tekintetben is vegyes eredményekhez vezettek. Van olyan tanulmány (Naps et al., 2000), amely szerint a diákok jobban teljesítenek, ha interaktív az animáció, de olyan kutatási eredményekkel is találkozhatunk (Jarc et al., 2000), amikor azok a tanulók teljesítettek jobban, akik úgymond zavartalanul el tudtak mélyülni a vizualizációban („nulla interaktivitás”). A *harmadik* kutatási kérdés, amelyet igyekszünk megválaszolni ebben a dolgozatban az, hogy létezik-e optimális interaktivitási szint, ha az algoritmusvizualizáció online történik.

Egy további kérdés, amely felmerül az online oktatási környezetekkel kapcsolatban, hogy mi a tanár szerepe ezekben a környezetekben. Számos visszajelzés arra mutat, olyan kollégáktól kaptunk, akik már évek óta használják az AlgoRythmics környezetet, hogy a diákok önállóan, csak részlegesen tudják kiaknázni a környezetben rejlő lehetőségeket. Egy érdekes idekapcsolódó

kérdés, hogy mennyire növelhető a tanulás minősége azzal, ha a tanár jól célzott kérdésekkel rávezeti a diákok figyelmét. A *negyedik* kutatásban éppen ezt vizsgáltuk.

Egy korábbi vizsgálat feltárta (Kátai, 2015), hogy az AlgoRhythmics tanulási környezet hatékony eszköz lehet humán érdeklődésű egyetemi hallgatók számítógépes gondolkodásnak a fejlesztéséhez is. Ezután felmerült a kérdés, hogy mennyire alkalmas arra, hogy különböző korosztályú tanulókat bevessünk a számítógépes algoritmusok világába. A jelen dolgozat *ötödik* fő pontja azt célozza meg, hogy 3., 5., 7. és 9. osztályos tanulók mennyire képesek elsajátítani egy alapvető informatikai algoritmust. E kutatás azt is vizsgálta, hogy a jelenlegi oktatási rendszer milyen mértékben járul hozzá a diákok számítógépes gondolkodása fejlesztéséhez.

1.3. TÉZISEK

A célkitűzések révén elővetített kutatások nyomán az alábbi téziseket sikerült bizonyítani:

1.3.1. Első kutatás tézisei

[T1.1] Azok a realisztikus algoritmusvizualizációk (mint például az AlgoRhythmics tánckoreográfiák), amelyek emberi mozgás révén illusztrálják az algoritmikai tartalmat, ugyanolyan hatékonyak, sőt hatékonyabbak is tudnak lenni, mint a megfelelő sematikus ábrázolások (absztrakt számítógépes animációk).

[T1.2] Könnyebb különbséget tenni a megtekintett algoritmusok stratégiái között, amennyiben ezeket sajátos realisztikusvizualizációk ábrázolták, mintha sematikus ábrázolások lettek volna alkalmazva.

[T1.3] Az olyan realiztikus algoritmusvizualizációk esetében, amelyek emberi mozgást alkalmaznak, főleg azok a figyelemelterelő elemek kerülendők, amelyek árnyékot vetnek azokra a mozgásokra, amelyek az algoritmus kulcsműveleteit illusztrálják (kevésbé jelentenek gondot az esetleges dekorációs figyelemelterelők).

1.3.2. Második kutatás tézisei

[T2.1] Amennyiben ismételt megtekintést alkalmazunk, az emberi mozgás effektust hasznosító realiztikus algoritmusvizualizációk (mint például az AlgoRhythms videók) hatékonyabbak, mint a megfelelő sematikusak ábrázolásai (például absztrakt számítógépes animációk).

[T2.2] Amennyiben sematikus és realiztikus algoritmusvizualizációt is alkalmazunk, előnyösebb lehet, ha előbb a sematikus, majd a realiztikus kerül megtekintésre. Mivel a realiztikus vizualizációk könnyebben felidézhetők, mint a sematikusak, ezért ismételt megtekintés, illetve több ábrázolás kombinálása esetén előnyös, ha a tanmenet realiztikus ábrázolásra végződik.

1.3.3. Harmadik kutatás tézisei

[T3.1] A előzetes programozási ismeretekkel nem rendelkező (kezdők – 0 év programozás) és kevés programozási tapasztalattal rendelkező (középhaladók – 1 vagy 2 év programozás) diákok javára válhat az algoritmus-vizualizáció interaktív bemutatása. A haladók (3 vagy 4 év programozás) otthonosan mozognak bármely interaktivitási szinttel rendelkező tanfolyam esetén.

[T3.2] Nincs szignifikáns különbség a különböző interaktivitási szintekkel tanuló fiúk és lányok eredményei között

[T3.3] Ha az interaktivitásból adódóan nagyon feldarabolódik az algoritmus-vizualizációs folyamat, akkor ez megnehezítheti az algoritmus stratégiájára való globális rálátást.

1.3.4. Negyedik kutatás tézise

[T4.1] A tanár által irányított kérdés-felelet megbeszélésnek döntő szerepe lehet az algoritmusvizualizációs környezetekben rejlő potenciál teljesebb kiaknázásában.

1.3.5. Ötödik kutatás tézisei

[T5.1] Ha egy oktatási rendszer nem fókuszál expliciten a számítógépes gondolkodás fejlesztésére, akkor a többi tantárgyak járulékos hozadékaként csak moderáltan fejlődik a képesség.

[T5.2] Megfelelő módszerekkel és eszközökkel már elemi szinten elsajátíthatók a számítógépes gondolkodás bizonyos elemei (ugyanazon hatékonysággal, mint felsőbb osztályokban), mindkét nem esetében.

1.4. Az értekezés áttekintése

Mivel kutatásaimat hangsúlyosan az AlgoRhythmics környezetben végeztem, ezért a 2. fejezetben röviden bemutatom az AlgoRhythmics projekt történetét, kiemelve, hogy mikor és hogyan kapcsolódtam be. A 3–7. fejezetek a dolgozatom gerincét képező öt kutatást fejtik ki. Az első négy kutatás a Sapientia EMTE Marosvásárhelyi Karának, elsőéves, reálszakos hallgatóinak bevonásával valósult meg a 2017/18-as, 2018/19-es és 2019/20-as tanévekben. A résztvevők 10–20%-a volt lány, ami jellemző az informatika és mérnöki szakokon tanulókra. Az ötödik vizsgálatban a marosvásárhelyi Bolyai Farkas Elméleti Líceum és a Művészeti Iskola diákjai vettek részt. Mindenik esetben a résztvevők beleegyezésüket adták adataik feldolgozásához. Ugyancsak közös

a kutatásokban, hogy a statisztikai elemzéseket az SPSS és R szoftvercsomagok egyikével végeztük, és 95%-os szignifikanciaszinttel ($p=0,05$) dolgoztunk. Az egyes csoportok varianciájának homogenitásvizsgálatára alkalmazott Levene próba értéke (F), valamint a hipotézisek kivizsgálása során számolt független t próba értéke (t) is értékes információkat nyújtottak számunkra.

Mivel legtöbb résztvevő számára még ismeretlen volt az AlgoRhythms környezet, ezért az 1–4 kísérletek elején tartottunk számukra egy rövid szemléltető bemutatót, a beszűrő rendezés tánc-koreográfiáját és animációját használva fel (ez az algoritmus nem szerepelt a mérésekben). Elmagyaráztuk, hogy mit értünk algoritmus alatt, illetve hogyan fogható fel egy összehasonlításra alapuló rendezési algoritmus úgy, mint „hasonlítás” és „hasonlítás + csere” műveletek sorozata. Azt is kihangsúlyoztuk, hogy a rendezési stratégiák több felvonásos forgatókönyveknek is tekinthetők: minden felvonásban/menetben helyére kerül egy vagy több elem. Mivel a kísérletekre a tanévek elején került sor, azt megelőzően, hogy elkezdtük volna a Programozás I tantárgyat, a programozói előismeretekkel nem rendelkező diákoknak ez volt az első kapcsolatuk a számítógépes algoritmusokkal.

Reméltük, hogy e felfezetővel mérsékeljük azt a jelenséget, amikor a tanulók ismeretlen ábrázolással bemutatott ismeretlen tartalmat kell, hogy elsajátítsanak („representation dilemma”) (Rau, 2017).

2. AZ ALGORHYTHMICS TANULÁSI KÖRNYEZET

Bizonyított tény (de Koning & Tabbers, 2011), hogy a nem hagyományos oktatási stratégiák bevezetésével a diákok fogékonyabbá válhatnak a tanulást illetően. Egy tánckoreográfia vagy egy dallam esetén, a művészi elemek által hozzáadott érték szárnyakat adhat a tanuláshoz. Egy gondosan megtervezett absztrakt vizualizáció segíthet elvonatkoztatni attól, ami a vizsgálat szempontjából lényegtelen, és a tanulmányozott algoritmus kulcsműveleteire fókuszálni. E két elem köré épül az AlgoRhythmic környezet: algoritmustáncok és számítógépes animációk a tanulók algoritmikus gondolkodása fejlesztése céljából.

Az AlgoRhythmic kutatócsoport, 2007-ben (Kátai et al., 2014), abból az ősi bölcsességből indult ki, hogy a tanulás annál hatékonyabb, minél több érzékszervet vonunk be ennek folyamatába. „Szükséges, hogy a megismerés mindig az érzékszervekből induljon ki (semmi sincs ugyanis az értelemben, ami nem volt meg előbb az érzékekben). Mi más ez, mint az, hogy a tanítás ne a dolgok szóbeli elbeszélésével vegye kezdetét, hanem a reális megfigyeléssel. És végül miután megmutattuk a dolgot, jöhet a bővebb magyarázat.” – valotta Comenius, a modern oktatás atyja (Pukánszky Béla, 1997). E comeniusi elvre építve, a kutatócsoport arra számított, hogy az általa bevezetett algoritmusillusztrációk révén könnyebb eljutni a megértésig, elsajátításig, rögzítésig, sőt mindezek járulékos hozadékaként fejlődik a tanulók számítógépes gondolkodása is.

Az AlgoRhythmic környezet tehát kezdetektől a többérzékszerves tanulás elvére épült. Az építkezés, lépésről-lépésre, érzékszervről-érzékszervre, modúlárisan történt. Mivel az aktív tanulás elvét is figyelembe vettük, a kezdeti ötlet egy többérzékszerves interaktív rendszerré fejlődött.

2.1. „Látni”, „hallani” és „tapintani” az algoritmusokat

A látás bevonása a tanítás-tanulásba elég kézenfekvő. Például, az oktatási anyagok régóta tartalmaznak képeket. Az algoritmusok viszont dinamikus folyamatok, ezért a szöveges leíráshoz társított statikus képeknél megfelelőbb ábrázolás lehet az animáció. Az utóbbi években bevett szokássá vált absztrakt animációk alkalmazása az algoritmusok oktatásánál.

Az AlgoRythmics kutatócsoport azonban ennél többet szeretett volna: bevonni a hallást és a tapintást is a programozás oktatásba. Mivel az algoritmusok meghatározó eleme a ciklus, az egyik alapötlet az volt, hogy megpróbálták „kihangosítani” az elemi algoritmusok ciklusszerkezetét (egy vagy több ciklusutasítás egymást követően, vagy egymásba ágyazottan). Ezen ciklusszerkezetek „megszólaltatásához” a kutatók kifejlesztettek egy szoftvereszközt (Kátai, 2014), amely hangjegysort társított az elemi algoritmusok kulcslépéseikhez. A diákoknak ezáltal lehetőségük volt meghallgatni és a hallás útján felismerni, ráérezni, az algoritmusok ciklusszerkezeteire.

Ami a tapintás bevonását illeti, az a megoldás született (Kátai, 2014), hogy az alkalmazás, amit kifejlesztettek, azt is lehetővé tette, hogy a felhasználó „bedobolja” az algoritmus ciklusszerkezetét a billentyűzetről (kiválasztott billentyűk által). Ez odavezetett, hogy a diákok érzékelték a ciklusok lüktetését, mintha lefogták volna az algoritmusok pulzusát. E modulnak köszönhetően nyert először a módszer interaktív jelleget is.

A módszer és a hozzá kapcsolódó eszköz hatékonyságának mérése érdekében a kutatócsoport több mérést is végzett informatika szakos középiskolás és egyetemista hallgatókon. A kutatási eredmények alátámasztották, hogy az érzékszervek (látás, hallás, tapintás) bevonása az informatikaoktatásba hatékonyabb tanulást eredményez (Kátai et al., 2008).

2.2. Algo-ritmika (eltáncolt algoritmusok)

Egy következő lépés az volt, hogy miként lehetne a kinesztetikus érzékelést is bevonni a tanulás folyamatába. Mivelhogy az algoritmus egy jól meghatározott műveletsort jelent, miért is ne lehetne felfogni úgy, mint egy forgatókönyvet, majd színre vinni. Ennek az ötletnek az implementálásaként a kutatócsoport koreográfiákat kidolgozott ki, melyek segítségével, maguk a diákok, tulajdonképpen „eljátszották” az algoritmusokat.

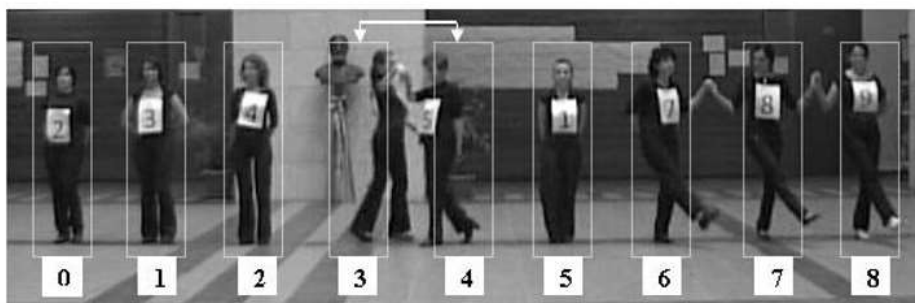
Mivel a keresésekről és rendezésekről szóló „kiselőadások” sikert arattak, a módszert rekurzív algoritmusokra is kiterjesztették. Az osztálytermi szerepalakítás volt tehát a megoldás arra, hogy a kutatók a mozgást is bevonják az algoritmusok oktatásába (1. ábra: Rekurzív függvény „eljátszva”).



1. ábra: Rekurzív függvény „eljátszva”

Ezzel az oktatási módszerrel tanítva újabb kutatásokat végeztek, amelyek során újra biztató eredmények születtek, megerősítve a csapatot abban, hogy jó az irány, amerre elindultak (Kátai, 2011).

Ekkor merült fel először az a kérdés, hogy a szerepalakítás helyett miért ne lehetne tánc, hogy újra megjelenjen a képben a hallás és a zene. Először amatőr táncosokat kértek fel, hogy Michel Flatley zenéjére rendezési algoritmusokat táncoljanak el (2. ábra: Buborékredezés). A videóra vett algoritmustáncok hatékonyságát az informatika oktatásában középiskolás diákokon mérték le (Kátai & Tóth, 2010).



2. ábra: Buborékredezés

Innen már csak egy lépés volt a ma már jól ismert AlgoRythmics táncokig. A kutatócsoport együttműködést kezdeményezett a Maros művészeggyüttessel (néptáncosokkal), akik segítségével a rendezési algoritmusokat illusztráló tánckoreográfiákat művészileg is színvonalassá tették. Ezen együttműködés gyümölcseként született meg 2011-ben hat rendezési algoritmus videóra vett tánckoreográfia illusztrációja (3. ábra: Rendezési algoritmusok videói).



3. ábra: Rendezési algoritmusok videói

A YouTube-on (www.youtube.com/user/AlgoRythmics) közzétett videók nézettsége, valamint az itt megosztott biztató megjegyzések (pl. „If my teacher had shown these videos, the class would have been clearer. Thank you so much!”), újabb szárnyakat adtak a környezet továbbfejlesztésére.

2.3. Egy webes alkalmazás az algoritmusok oktatásához

A kutatók kezdetektől tisztában voltak azzal, hogy bár a tánc-koreográfia ábrázolásoknak meg lehetnek a maguk előnyei, nem helyettesítik az absztrakt animációkat. Az absztrakt ábrázolások vitathatatlan előnye, hogy elvonatkoztatnak a lényegtelen részletektől, és segítenek az algoritmusok kulcselemeire összpontosítani. Következésképpen a csoport létrehozta az AlgoRythmics webes környezetet, ahol minden videó mellé társítva lett egy absztrakt animáció. Ezen új modul egy további erőssége az volt, hogy interaktív üzemmód-

ban is működött. Lehetővé tette a megtekintett animációk interaktív rekonstruálását (ugyanazon a számsorozaton), illetve azok levezénylését véletlen számsorozatokon. A levezénylés két üzemmódban is működött: amikor látszottak a számok, és amikor el volt rejtve a számsorozat.

A kiteljesedett AlgoRythmics módszer és a kapcsolódó webes alkalmazás, 2013-ban elnyerte az Informatics Europe, „Best Practice in Education” díját.

Az új eszköz további kutatásokat tett lehetővé. Például bizonyítást nyert, hogy a diákok jobban teljesítenek, ha rejtett számsorozatokon tanulmányozzák az algoritmusokat. Ennek egy lehetséges magyarázata, hogy az úgynevezett „szelektív elrejtés” elvének ily módú alkalmazása segíti a diákokat abban, hogy könnyebben azonosuljanak a „vak számítógép nézőpontjával” (Káta, 2014). E tanulságot úgy kamatoztattuk a későbbi fejlesztéseknél, hogy néhány újabb tánccoreográfiában a szereplők csak a hátukon viselik a számokat (szemből nézve őket, a számsorozat rejtett).

2.4. Algoritmika mindenkinek

Végül az AlgoRythmics alkalmazás több lett, mint többérvékszerves interaktív tanulási környezet. Talán legszembetűnőbb sajátossága, hogy egyedi módon kombinálja a tudományt a művészetekkel. Ennek tudatosulása vezetett el ahhoz a kezdeményezéshez, hogy vizsgáljuk a módszer és eszköz hatékonyságát nem informatika szakos hallgatókon is. Az eredmények nagyon biztatóak voltak (Káta & Adorjáni, 2014; Káta, 2015).

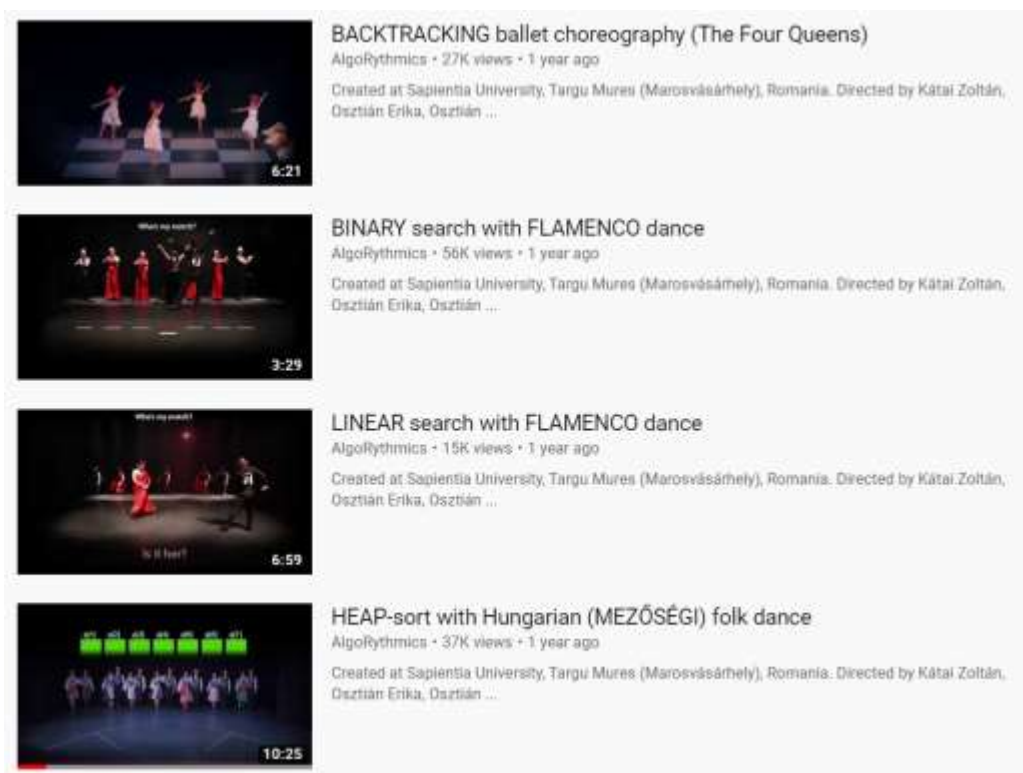
Ami az alkalmazott tánc típusokat illeti, a kutatók az Erdélyben honos etnikumok néptáncjaiból merítettek: magyar, székely, román, cigány, szász, csángó. Mivel mindezekhez, természetesen, nemzeti érzület is kapcsolódik, egy olyan kutatás is született, amely azt vizsgálta, hogy befolyásolja-e mindez a tanulás folyamatát. Érdekes eredmények születtek, amikor román, magyar és cigány

néptáncok által illusztrált algoritmusokat alkalmaztak tiszta magyar, tiszta román és vegyes oktatási környezetben (Kátai, 2014b).

2.5. Az AlgoRhythmics projekt új lendületet kap

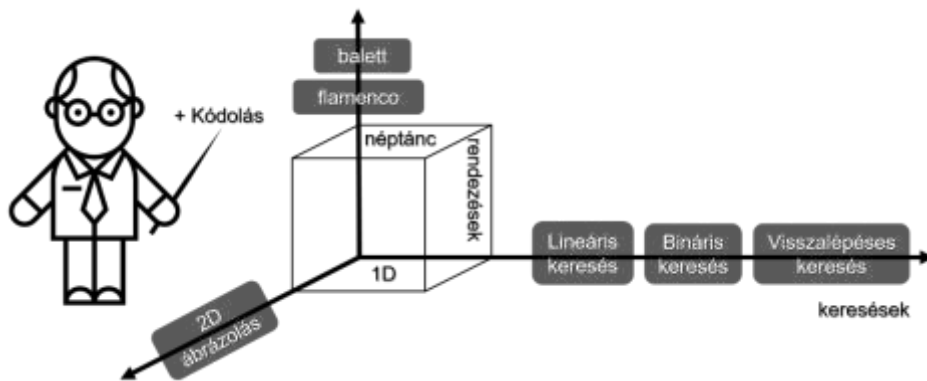
2016-tól az AlgoRhythmics projekt új lendületet kapott. Ekkor csatlakoztam jómagam is a kutatócsoporthoz. Döntés született, hogy a projektet az alábbi négy dimenzió mentén fejlesszük tovább: (1) algoritmus típus; (2) tánc típus; (3) ábrázolási mód; (4) táncról a kódig.

1. Kiterjesztettük az algoritmus-kollekciót a keresési stratégiákra is: lineáris keresés, bináris keresés, visszalépéses módszer (backtracking) (4. ábra: Új algoritmusok).



4. ábra: Új algoritmusok 2016-tól

2. A számsorozatok egydimenziós megjelenítése kiegészült kétdimenziós ábrázolásokkal; például az újonnan elkészült heap-sort koreográfiában az egydimenziós tömbként ábrázolt kupac bináris fává nyílik ki.
3. Nyitottunk újabb táncstílusok fele: flamenco és balett.
4. Teljesen újraterveztük a webes alkalmazást, amely olyan modulokkal bővült, amelyek elvezetik a diákokat az algoritmusok megértésétől ezek implementálásáig (5. ábra: AlgoRythmics új dimenziói).



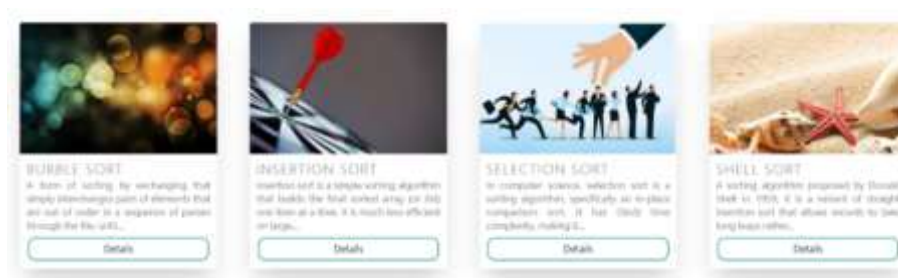
5. ábra: AlgoRythmics új dimenziói

Mivel az általam végzett kutatások hangsúlyosan ebben a megújult környezetben zajlottak, ezért az alábbiakban bemutatom ennek néhány elemét. Munkámat nagymértékben elősegítette az eszköz kettős jellege. Egyrészt egy olyan tanulási környezetről van szó, amely segíti a diákokat alapvető számítógépes algoritmusok tanulmányozásában. Másrészt viszont, mivel a felhasználói tevékenység naplózásra kerül, az eszköz kiválóan alkalmas kutatási vizsgálatok megvalósítására is. A felhasználói visszajelzések azért is hasznosak voltak, mert irányt mutattak a további fejlesztésekhez.

Az AlgoRythmics élményt a következőképpen tudnám körülírni. Egy algoritmus felfogható forgatókönyvként, amelyet a tánc vagy animáció, úgymond

színre visz. A mi esetünkben e „jelenetek” helyszíne az AlgoRhythmics környezet. Minden idelátogató kiválaszthatja, hogy melyik jelenetet (algoritmust) szeretné megtekinteni, milyen ábrázolási módban, illetve mennyire szeretne bekapcsolódni (beállíthatja az interaktivitás szintjét). Amint a tanuló „elkapja” az algoritmus ritmusát, bekapcsolódhat a kódépítésbe. A környezet mintegy kézen fogva segíti át a diákot a megértett algoritmus és ennek számítógépes kódja között tátongó szakadékon. Amint megszületett a felhasználói kód, úgymond „életre is kel”, abban az értelemben, hogy most már ez vezérli a háttérből az animációt.

A kiegészült AlgoRhythmics kollekció tíz algoritmust foglal magába: beszűrő, kiválasztó, buborék, shell, összefésülő, gyors és kupac rendezés, valamint lineáris-, bináris- és visszalépéses-keresés (6. ábra: Rendezési algoritmusok). A környezet tanulási alapelemei, a tanulási lépések (7. ábra).



6. ábra: Rendezési algoritmusok



7. ábra: Tanulási lépések listája

- a kiválasztott algoritmus videó-illusztrációjának (tánc-koreográfia) megtekintése
- a kiválasztott algoritmus számítógépes animációjának megtekintése
- a kiválasztott algoritmus interaktív levezénylése
- a számítógépes kód interaktív megépítése
- életre kel a kód (annak szimulálása, hogy a felhasználói kód vezérli az animációt)

E tanulási lépések legtöbbjénél az alábbi paraméterek állíthatók be:

- a bemenet típusa
 - véletlen számsorozat
 - tanár által megadott számsorozat
 - legjobb esetnek („best-case”) megfelelő számsorozat (rendezések esetén növekvő)
 - legrosszabb esetnek („worst-case”) megfelelő számsorozat (rendezések esetén csökkenő)
- a lejátszás típusa
 - automatikus (zavartalan megfigyelést tesz lehetővé)
 - interaktív (a vizualizáció kulcs momentumokban megáll, és a folytatás végett rákérdez a felhasználóra)
- megjelenítés típusa
 - a számsorozat látszik („white-üzemmód”)
 - a számsorozat el van rejtve („black-üzemmód”)

A tanulási lépésekből tanfolyamok rakható össze. Minden algoritmushoz csupán egy alapkurszus rendelhető, azonban specifikus tanfolyamokból, bármikor, bármennyi társítható hozzá.

A környezet lehetővé teszi a tanulói teljesítmény mérését tesztek révén (8. ábra: Kiértékelő tesztek). Az eredmények pedig meg is jeleníthetők (9. ábra: Profil oldal).



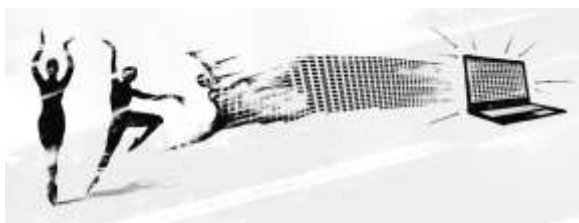
8. ábra: Kiértékelő tesztek



9. ábra: Profil oldal

Összegezve elmondható, hogy az AlgoRythmics környezet valóban egy interaktív, többérzékszerves, elektronikus oktatási környezetté nőtte ki magát, amely kiválóan alkalmas arra, hogy bevezesse az idelátogatókat a számítógépes algoritmusok világába, illetve hozzájáruljon ezek számítógépes gondolkodásának fejlődéséhez, eljuttatva őket a táncról a kódig (10. ábra: A táncról

a kódig). Ráadásul az AlgoRythmics platform nem csak tanulási környezet, hanem jól használható kutatási eszközként is.



10. ábra: A táncról a kódig

Az értekezésemet 5 nagy gondolatra építettem, amelyek a legfrissebb kutatásaink eredményeiből fogalmazódtak meg. Ezeket közlésre az alábbi folyóiratokhoz nyújtottuk be:

Schematic and human movement effect enhanced realistic algorithm visualization in the context of AlgoRythmics learning environment

Acta Polytechnica Hungarica, Journal of Applied Sciences

Schematic versus human movement effect enriched realistic dynamic visualization

Journal of Computer Assisted Learning

Algorithm visualization environments: can an optimal interactivity level be established?

Frontiers in Education

Improving AlgoRythmics Teaching Learning Environment by Asking Questions

International Journal of Instruction

Investigating young school students' computational thinking ability across grade levels

Journal of Educational Computing Research

3. SEMATIKUS VERSUS EMBERI MOZGÁST ALKALMAZÓ REALISZTIKUS ALGORITMUSVIZUALIZÁCIÓ

3.1. Szakirodalmi áttekintés

3.1.1. Sematikus versus dinamikus algoritmusvizualizációk

Amint már említettük, a számítógépes algoritmusoknak kiemelt szerepük van a tanulók számítógépes gondolkodásának fejlesztésében. Mivel az algoritmusok elvont folyamatok, a vizualizáció egy kritikusan fontos eszköz lehet a hatékony oktatásukhoz. Mivel az algoritmusok természetükben dinamikusak, ezért a legkézenfekvőbb, hogy dinamikus vizualizációkat alkalmazzunk az illusztrálásukhoz, mint amilyenek az animációk és a videók.

Általában, egy dinamikus vizualizáció lehet: sematikus vagy realisztikus. (Nugteren et al., 2014). Ebben a kutatásban arra fókuszáltunk, hogy felmérjük e két ábrázolási mód egymáshoz viszonyított értékét (a vizsgált tartalom megértéséhez való hozzájárulásuk tekintetében) az algoritmusvizualizáció témakörben.

Több kutatás (Byrne, 2001) is összehasonlította már a sematikus és realisztikus vizualizációkat, de egyikük sem az algoritmusvizualizáció témakörre összpontosított. Egy lehetséges ok, amiért ez egy kevésbé kutatott terület, hogy nem kézenfekvő az algoritmusok realisztikus ábrázolása. Mivel az algoritmusok elvont folyamatok, megszokottá vált, hogy sematikus ábrázolásokkal, azaz absztrakt animációk által illusztrálják működésüket. Másfelől viszont születtek olyan vizualizációk (Castro-Alonso et al., 2014) is, főként az unplugged informatikaoktatás keretében, amelyek közelebb állnak a realisztikus ábrázoláshoz. Egy ilyen példa az AlgoRhythmics (2020) környezet is,

amely realisztikus tánckoreográfiákkal illusztrál számítógépes algoritmusokat.

Az eddigi kutatási eredmények azt mutatják, hogy mind a kétfajta ábrázolási módnak meg lehetnek az előnyei attól függően, hogy mi a szemelőtt tartott tanulási cél. Például a sematikus ábrázolások egyik fontos előnye, hogy segítenek a tanulóknak abban, hogy a tanulás szempontjából lényeges elemekre összpontosítsanak, illetve, hogy el tudjanak tekinteni a lényegtelen elemektől. Másfelől azonban, ahogy ezt Goldstone és Son (2005) kutatók is hangsúlyozzák, a realisztikus ábrázolásoknak az lehet az előnyük, hogy könnyebb visszaemlékezni rájuk. Emellett azzal a tulajdonsággal is rendelkeznek, hogy növelhetik a tanulók motiváltságát, merthogy, például, izgalmasabbak, érdekesebbek tudnak lenni, mint a sematikus ábrázolások.

3.1.2. Emberi mozgás effektus

Egy meglepő kutatási eredmény a statikus versus dinamikus vizualizáció témakörben, hogy hatékonyságuk tekintetében vegyes eredmények születtek (Ainsworth & VanLabeke, 2004). Vannak olyan kutatások, amelyek úgy következtetnek, hogy a dinamikus vizualizációk a hatékonyabbak (Goldstone, & Son, 2005), mások viszont a statikus ábrázolásokat találták előnyösebbnek (Butcher, 2006) és vannak olyanok is (Scheiter, 2009), amelyek arra jutottak, hogy nincs szignifikáns különbség a kettő alkalmazása között. Az egyik lehetséges ok, amiért a dinamikus vizualizációk nem következetesen hatékonyabbak, ezek tranziens jellegéből adódhat. Amíg egy statikus kép tanulmányozásában elmélyülhet a tanuló, addig a dinamikus vizualizációk során egymásra tevődik az éppen megtekintett jelenet processzálása, beépítése és az új jelenet észlelése.

Érdekes azonban, hogy friss kutatások feltárták, hogy abban az esetben, ha a dinamikus vizualizáció emberi mozgás által ábrázolja a tanulmányozandó folyamatot, a dinamikus ábrázolásokkal való tanulás következetesen hatékonyabb, mintha ez statikus képek segítségével történne (Castro-Alonso et al., 2014). Úgy tűnik, hogy az emberi elme annyira emberi mozgásra orientált, hogy amikor ilyen mozgást kell követnie, a tranziens effektből származó hátrány eltűnik vagy minimálisra csökken. A szakirodalom ezt a jelenséget „emberi mozgás effektusnak” („human movement effect”) nevezi (Shaffer et al., 2010).

Bár mind a realisztikus, mint a sematikus ábrázolásoknak meg vannak az előnyeik, azért a kutatások többsége a sematikus vizualizációk javára következtetnek. Mint releváns példát megemlíthetjük Scheiter és munkatársai (2010) cikkét, akik azt vizsgálták (biológiaoktatás keretében), hogy a sematikus és realisztikus vizualizációk milyen kombinációban a leghatékonyabbak. A négy csoportra felosztott tanulók különböző tanmenetek alapján tanultak: az első csoport 2-szer tekintette meg ugyanazt a realisztikus ábrázolást, a második csoport 2-szer ugyanazt a sematikus ábrázolást, a harmadik csoport ugyanannak a jelenségnek először a sematikus, majd a realisztikus ábrázolását, a negyedik pedig ugyanannak a jelenségnek először a realisztikus, majd a sematikus ábrázolását tanulmányozta. Az eredmények azt mutatták, hogy akik kétszer tekintették meg ugyanazt a realisztikus ábrázolást szignifikánsan alulteljesítettek a másik három csoporthoz képest. Vagyis a kutatók úgy találták, hogy a realisztikus vizualizáció kevésbé volt hatékony, mint a sematikus.

Egy hasonló, frissebb kutatásban (Nugteren et al., 2014) ugyancsak négy csoportot vizsgáltak, a következő tanmenetek szerint: az egyik csoport csak a sematikus ábrázolást tekintett meg, a második csak a realisztikust, a harmadik csoport először a sematikust majd a realisztikust, egymást követően, a negyedik pedig párhuzamosan követhette, a képernyő baloldalán a sematikus, a

jobboldalán pedig a realisztikus vizualizációt. Az eredmények megint csak azt mutatták, hogy az a csoport teljesített a leggyengébben, amely csak realisztikus ábrázolással tanult.

E következtetések összecsengenek a Tversky, Morrison és Betrancourt (2002) kutatók kijelentésével, akik azt hangsúlyozzák, hogy az animációk a lehető legkevesebb realisztikus elemet kellene, hogy tartalmazzanak, mert még az úgynevezett izgalmas realisztikus részletek is akadályozzák a dinamikus vizualizációban megjelenített mozgás interpretálását. Ennek egy lehetséges oka, hogy a dinamikus ábrázolásokban gyakran éppen a mozgás hordozza az elsajátítandó információt.

Feltevődik azonban az a kérdés, hogy mi van akkor, ha a realisztikus ábrázolás emberi mozgást alkalmaz? Ebben a kutatásban, többek között erre a kérdésre kerestük a választ, levetítve azt az AlgoRhythmics környezetre, ahol a realisztikus ábrázolásokat a tánckoreográfiák, a sematikusakat pedig a számítógépes animációk képviselik.

3.1.3. „Egyszínű animációk” versus „sokszínű táncok”

Az AlgoRhythmics környezetbe beépített absztrakt animációk egy közös vonása, hogy mindeniken ugyanazok a motívumok vonulnak végig, úgymond „egyszínűek”. Mindenik vizualizációban ott van a tömb, amely hasonló módon van ábrázolva, a kulcsműveletek, az összehasonlítás és a csere, amelyek ugyanúgy vannak animálva. Továbbá, például a rendezési algoritmusok esetén, azok az elemek, amelyek a helyükre kerültek, szint váltanak.

Ezzel szemben mindenik tánckoreográfiának egyedi stílusa van, amely a felhasznált tánc típusok sokszínűségében gyökerezik. Persze a videókban is vannak közös jegyek. Például a számsorozatot, amelyen a rendezés vagy keresés zajlik, mindig egy táncos-sorozat ábrázolja. A lineáris és bináris keresések

esetében a tömböt, amely tárolja a számsorozatot, a székek ábrázolják, amelyeken a lányok ülnek. A rendezési algoritmusok esetében a tömböt ábrázoló „rekeszeket” utólag vittük rá a videóra, videószerkesztő szoftver segítségével. A rendezési táncokban a helyükre került elemeket ábrázoló táncosok gyakran megfordulnak. Másfelől viszont, mindenik videóban egyedi az, ahogy a két kulcsművelet, az összehasonlítás és a csere eltáncolásra kerül, ugyanis ennek módja igazodik a megfelelő táncstílushoz.

Az a tény, hogy mindegyik videónak, úgymond megvan a maga egyénisége, hozzájárulhat ahhoz, ami általában is igaz a realisztikus ábrázolásokra, hogy könnyebb visszaemlékezni rájuk. Ennek egyenes következménye lehet, hogy azok a diákok, akik a tánccoreográfiákkal tanulták az algoritmusokat, a visszaemlékezésükön könnyebben különbséget tudnak tenni a vizsgált algoritmusok között. E kutatásban ezt a jelenséget is vizsgáltuk.

3.1.4. Dekorációs versus reprezentációs figyelemelterelő elemek

Höffler és Leutner (2007) különbséget tesz egy vizualizáció reprezentációs, illetve dekorációs elemei között. Reprezentációs elemnek számít az, amelyik a tanulás tárgyát képező jelenség ábrázolására szolgál. Ezzel szembe a dekorációs elemek elsődleges szerepe az, hogy motivációt gerjesszenek. A tanulmányok arra mutatnak, hogy akkor hatékonyabbak a dinamikus vizualizációk, mint a statikus képek, ha az animált elemek reprezentációs természetűek nem pedig dekoratívak. E terminológia alapján két osztályba sorolhatók az AlgoRhythmic videók realisztikus elemei is. Dekorációs művészi elemeknek számítanak, például, a ruhák, a zene, a férfi és női táncosok. A sajátos tánclépések viszont, amelyek az algoritmusok kulcsműveleteit ábrázolják, reprezentációs művészi elemekként foghatók fel. Ezzel összhangban, a potenciálisan figyelemelterelő elemeket is két kategóriába sorolhatjuk: dekorációs

figyelemelterelők és reprezentációs figyelemelterelők. A dekorációs művészi elemek, nyilván, egyben potenciális figyelemelterelők is. A reprezentációs művészi elemek akkor számítanak figyelemelterelőnek, ha árnyékot vetnek az algoritmus műveletei és az ezt ábrázoló mozgás közötti egy-egy megfeleltetési kapcsolatra.

Különösen érdekeltek voltunk abban, hogy mennyire érinti ez a kétfajta figyelemelterelő elem az AlgoRhythmics videók hatékonyságát. Ezért a kísérlethez két algoritmust választottunk: egy olyat, amely esetében a tánckoreográfia csak dekorációs természetű potenciális-figyelemelterelőket tartalmaz (kiválasztó rendezés), és egy olyat, amely esetében jelen vannak a reprezentációs jellegű potenciális figyelemelterelők is (buborékrendezés).

3.2. Kutatási kérdések

A korábbi kutatások, amelyek az AlgoRhythmics környezetet vizsgálták, ami a tánckoreográfiákat illeti, főleg ezek motivációs értékét elemezték. Ebben a kutatásban azonban arra fektettünk hangsúlyt, hogy milyen mértékben járul hozzá a kétféle ábrázolás az algoritmusok megértéséhez. Ennek érdekében a megtervezett tanmenet két algoritmus (buborék és kiválasztó rendezés) számítógépes animációja (sematikus ábrázolás) és tánckoreográfia vizualizációja (realisztikus ábrázolás) köré épült.

A fenti szakirodalmi áttekintés nyomán az alábbi kutatási kérdéseket fogalmaztuk meg.

1. Tud-e ugyanolyan hatékony lenni egy realisztikus dinamikus ábrázolás, mint a megfelelő sematikus vizualizáció, amennyiben emberi mozgást alkalmaz?

2. Könnyebb-e visszaemlékezni, különbséget tenni a megtekintett algoritmusvizualizációk között, amennyiben ezeket realisztikus tánckoreográfiák ábrázolták, mintha az illusztrálásuk sematikus ábrázolások révén történt volna?
3. Hogyan befolyásolják a realisztikus vizualizációkkal való tanulás hatékonyságát a dekorációs, illetve reprezentációs figyelemelterelő elemek?

3.3. KÍSÉRLET

A kísérletet háromszakaszra terveztük: előteszt, tanulási szakasz, utóteszt. A kísérletre a 2019/20-as tanév elején került sor, az első héten az előtesztre, a másodikon a tanulási szakaszra és az utótesztre.

3.3.1. A kutatás résztvevői

Az előre meghirdetett felmérésre meghívtunk minden elsőéves, reálszakos hallgatót. Az előteszten 137 diák vett részt az informatika, számítástechnika, automatizálás, távközlés, mechatronika és gépészmérnöki szakokról, akiket véletlenszerűen osztottunk két csoportba: kontroll és kísérleti. Az utóteszt, illetve adattisztítás után kiderült, hogy végül 116 olyan hallgató volt (14,65% lány), akik mindkét teszten jelen voltak, és a feladatokat végig oldották. Őket tekintettük a kutatás kiértékelhető résztvevőinek. Úgy sikerült, hogy a 116 résztvevőnek éppen a fele (58 diák) volt kontroll csoportbeli, és a másik fele (58 diák) tartozott a kísérleti csoporthoz. Azokat a hallgatókat tekintettük kontroll csoportnak, akik (s)ematikus ábrázolással (absztrakt számítógépes animációkkal) tekinthették majd meg a kiválasztott algoritmusokat (S-csoport). Mivel a környezet újszerűsége főleg a tánckoreográfiákban rejlik, ezért azok a hallgatók képezték a kísérleti csoportot, akiknél a (r)ealisztikus ábrázolásokat alkalmaztuk (R-csoport).

3.3.2. A kutatás megvalósításának eszközei

Mivel a résztvevők csoportja vegyes volt, ami a programozói előismereteket illeti, ezért az előteszt célja az volt, hogy olyan feladatokon mérjük le a résztvevők számítógépes gondolkodását, amelyek nem feltételeznek programozói előismereteket. Ehhez a teszthez a feladatokat a jól ismert Bebras verseny magyarországi megfelelőjének (e-hód) adatbázisából válogattuk (A függelék: Számítógépes gondolkodás teszt).

Az előteszt felépítését a Mellékletek 10. táblázat: Kérdőív szerkezete mutatja be. A tanulási szakaszhoz a buborék és kiválasztó rendezések, AlgoRythmics környezetbeli sematikus és realisztikus vizualizációit választottuk. Mindkét algoritmus úgy fogható fel, mint „hasonlít” és „hasonlít+csere” műveletek jól meghatározott sorozata. Azért esett éppen erre a két algoritmusra a választás, mert a kiválasztó rendezés videójában a „hasonlít” -táncjelenetek ugyanazzal a tánclépéssel vannak illusztrálva, függetlenül attól, hogy különálló „hasonlít” műveletről van szó, vagy „hasonlít+csere” párosról. Másfelől viszont, a buborékrendezést illusztráló videóban a kombinált „hasonlít+csere” jelenetekben, nincsenek különálló „hasonlít” és „csere” szakaszok. Ezen sajátossága miatt a buborékrendezés tánckoreográfiát úgy tekintettük, mint amely nemcsak dekoratív, hanem reprezentációs figyelemelterelő elemeket is tartalmaz.

Az utóteszt kérdései (Mellékletek 10. táblázat) azt mérték fel, hogy a résztvevők mennyire értették meg a két algoritmust. A kérdések megválaszolásához a hallgatók el kellett, hogy képzeljék a megtekintett algoritmusok működését egy véletlen számsorozaton, illetve legjobb (növekvő számsorozaton) és legrosszabb (csökkenő számsorozaton) esetekben.

A kísérlet implementálásának részletei végett lásd a melléklet 1. kutatásra vonatkozó részét.

3.4. Eredmények és kiértékelés

	Nem		Hány évet tanult programozást?			Tanultad a buborékrendezést?		Tanultad a kiválasztó rendezés?	
	fiú	lány	0	1–2	4	igen	nem	igen	nem
R-csoport	86%	14%	30%	22%	48%	55%	45%	29%	71%
S-csoport	84%	16%	40%	29%	31%	47%	53%	19%	81%
Fisher-egzakt teszt	(2×2) $p = 1 > 0,05$		(2×3; Freeman-Halton extension) $p = 0,18 > 0,05$			(2×2) $p = 0,45 > 0,05$		(2×2) $p = 0,27 > 0,05$	

1. táblázat: Fisher-féle egzakt tesztjének eredménye

A két csoport összetételére vonatkozó adatok összehasonlításánál a Fisher-egzakt tesztet használtuk, hisz a vizsgált változók kategoriális típusúak voltak (fiú/lány, 0/1–2/4 év, igen/ nem). Amint az 2. táblázat-ból is kiderül, ezen adatok alapján, a két csoport statisztikailag ekvivalens volt (az összes p-érték nagyobb, mint a szignifikancia szint).

A két csoport előteszt teljesítményének az összehasonlításához varianciaanalízist (ANOVA) használtunk. Ami az utóteszt eredményeket illeti, ezek vizsgálatához a variancia-kovariancia analízist (ANCOVA) választottuk. E tesztek használatának egyik előfeltétele, hogy az összehasonlítandó csoportok szóráshomogenitása egyenlő legyen. Ennek ellenőrzése végett Levene-próbát alkalmaztunk, amely minden esetben szignifikancia szint feletti p-értéket eredményezett: az előteszt kapcsán mindkét rendezésnél $p=0,8>0,05$, az utóteszt esetén a buborékrendezésnél $p=0,07>0,05$ a kiválasztó rendezésnél pedig $p=0,72>0,05$ értéket kaptunk.

Az előteszt eredmények elemzésénél a független változó a tanmenet típusa volt (realisztikus vagy sematikus), a függő változó pedig a résztvevők előteszt teljesítménye (a számítógépes gondolkodásukra vonatkozó kérdésekre adott válaszaik alapján). Az eredmények azt mutatták, hogy a két csoport e

tekintetben is statisztikailag egyenértékűnek tekinthető (R-csoport: 68%, S-csoport: 69%; $F(1,114) = 0,1$, $p = 0,74 > 0,05$).

A diákok utóteszt válaszait a következőképpen kódoltuk (mindkét algoritmus kapcsán a 0–13 intervallumból kaphattak pontokat).

- Az első 4 kérdés válaszaire (1–4 kérdések a buborékrendezésnél, vagy 5–8 kérdések a kiválasztó rendezésnél) 4 pont helyett 2-t kaptak azok a diákok, akik következetesen helyesen jelölték meg az elempárokat, amelyekre az algoritmus első 4 művelete vonatkozott, viszont az elempárokhoz helytelen műveleteket társítottak.
- A 9, 12, 20 és 10, 16, 24 kérdések esetén a helyes válaszokra 1, a helytelen válaszokra 0 pontot kaptak.
- A 11, 13, 14 és 15, 17, 18 kérdések esetén 0,75 pontot kaptak 1 pont helyett azok, akik 4 helyett 5-öt, vagy 10 helyet 15-öt válaszoltak (azaz $n=5$ esetén $n-1$ helyett n , illetve $n(n-1)/2$ helyett $n(n+1)/2$ képlettel dolgoztak).
- A 19, 21, 22 és 23, 25, 26 kérdések esetében hasonlóképpen 0,75 ponttal voltak osztályozva a résztvevők, ha $n-1$ helyett az n , illetve $n(n-1)/2$ helyett az $n(n+1)/2$ képletet adták meg. Azok a diákok, akik rossz képletet adtak meg, de a megadott polinomnak a fokszáma helyes volt (lineáris vagy négyzetes) 0,25 pontot kaptak.

3.4.1. Absztrakt animációk versus tánckoreográfiák

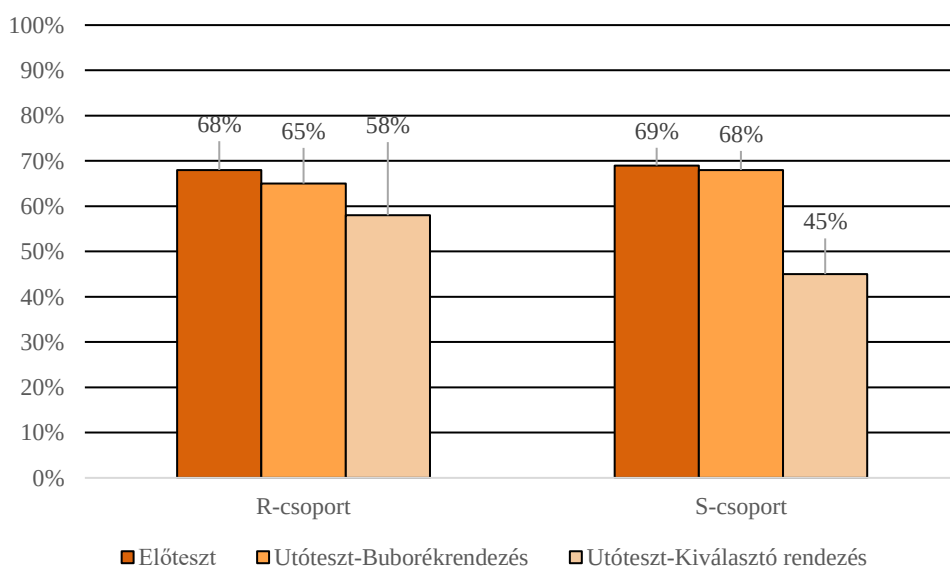
Az utóteszt eredményeinek az összehasonlítására az ANCOVA tesztet használtuk. A független változó újra a tanmenet típusa volt, a függő változó pedig az utóteszt kérdésekre adott válaszok pontszámai. Az előteszt eredményeket, mint kovariáns használtuk. Az elemzés azt mutatta, hogy nincs szignifikáns különbség a két csoport között a buborékrendezésre vonatkozóan

($F(1,113)=0,78$, $p=0,37>0,05$). A kiválasztó rendezés esetében viszont a kísérleti csoport (R-csoport) szignifikánsan jobban teljesített, mint a kontroll csoport (S-csoport) ($F(1,113)=9,52$, $p=0,002<0,05$; partial $\eta^2 = 0,07$) (2. táblázat). Az ANCOVA teszt is azt mutatta, hogy ezt az eredményt nem befolyásolta az előtesztnél tapasztalt minimális (nem szignifikáns) különbség.

Előteszt		Utóteszt	
		Buborékrendezés	Kiválasztó rendezés
R-csoport	68%	65%	58%
S-csoport	69%	68%	45%

2. táblázat: Elő- és utóteszt eredmények a két csoport szignifikancia különbségének mérésére

Az adatok elemzéséből az is kiderült, hogy mindkét csoport jelentősen jobban teljesített az első algoritmuson, mint a másodikon (R-csoport: 65% vs. 58%; S-csoport: 68% vs. 45%)



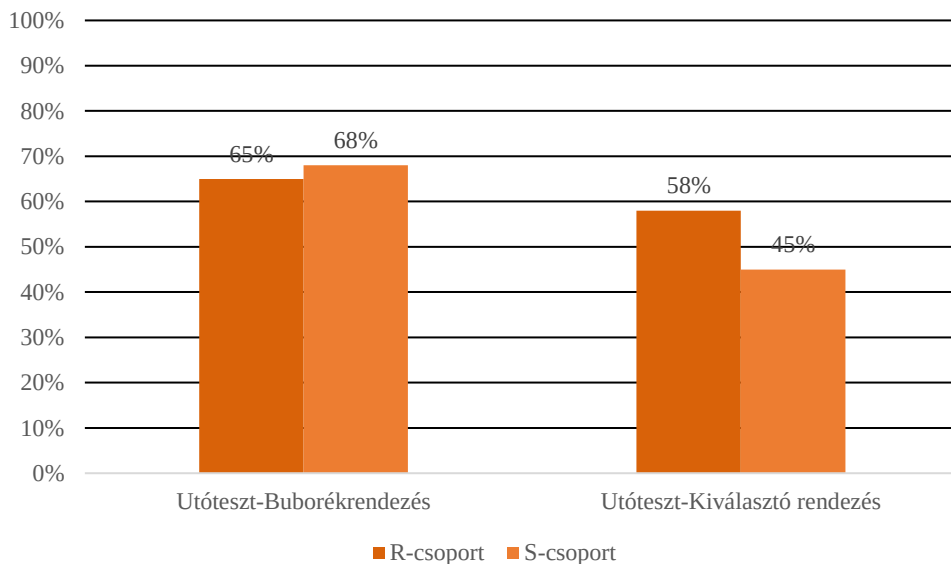
11. ábra: A két csoport előteszt és utóteszt eredményei

Ez várható is volt, hiszen lényegesen több résztvevő válaszolta azt, hogy már ismerte a buborékrendezést, mint ahányan a kiválasztó rendezést (bár az összes diák részére, akik tanultak programozást a középiskolában, a tanterv mindkét algoritmust tartalmazta).

Az első kutatási kérdésünkkel kapcsolatban az eredmények arra mutatnak, hogy az R-csoport tagjai nem teljesítettek alul az S-csoportéhoz képest (11. ábra)

Igaz, hogy ez a következtetés ellentétben áll néhány korábbi kutatás eredményeivel, amelyek a sematikus ábrázolások javára következtetnek, viszont összhangban van azokkal a friss eredményekkel (Castro-Alonso, 2014), amelyek az emberi mozgás értékét hangsúlyozzák dinamikus ábrázolások esetében. Más szóval, egy lehetséges magyarázat az R-csoport teljesítményére az lehet, hogy tagjai olyan realisztikus ábrázolással tanulták az algoritmusokat, amelyek hasznosítják az emberi mozgás effektust.

Annak magyarázata mögött, hogy a második algoritmus esetében, nemcsak, hogy nem maradt alul az R-csoport, hanem szignifikánsan jobban teljesített, egy összetettebb jelenség állhat.



12. ábra: Buborék vs. Kiválasztó rendezés eredményei az utóteszten

Úgy tűnik, hogy mivel mindkét csoport a két algoritmust közvetlen egymás után tekintette meg, ez kissé megzavarhatta a diákokat. Ahogy már említettük, korábbi kutatások úgy következtetnek, hogy a realisztikus vizualizációk egyik előnye az lehet, hogy könnyebb visszaemlékezni rájuk (Goldstone & Son, 2005). Az AlgoRythmics videók esetében, a sajátos művészi elemeknek köszönhetően, ez különösképpen igaz lehet. Ahhoz, hogy válaszolni tudjanak az utóteszt kérdéseire, amelyek a két algoritmusra vonatkozóan alternálva jelenetnek meg a kérdőíven, a diákoknak világosan külön kellett választani elméjükben a két rendezési stratégiát. A második kutatási kérdésünkkel kapcsolatban tehát, úgy következtethetünk, hogy az R-csoport jobb teljesítményéhez az járulhatott hozzá, hogy azok a diákok, akik realisztikus ábrázolással tekintették meg az algoritmusokat, jobban vissza tudtak emlékezni ezekre, illetve könnyebben különbséget tudtak tenni közöttük.

Egy következő lépésként kérdés, illetve kérdéscsoportok szerinti bontásban is megvizsgáltuk az eredményeket. A buborékrendeztés esetében szignifikáns

különbség volt a két csoport között a következő kérdéseknél (mindenik esetben az S-csoport teljesített jobban, mint az R-csoport, kivéve a 9. kérdést):

- A 1–4 kérdések, az első 4 művelet, $p=0,01<0,05$
- A 9. kérdés, a második menet első művelete, $p=0,03<0,05$
- A 11. kérdés, a hasonlítás műveletek száma 5 hosszú számsorozat esetén, $p=0,01<0,05$
- A 19. kérdés, a hasonlítás műveletek száma n elemű számsorozat esetén, $p=0,01<0,05$

A kiválasztó rendezés esetében szignifikáns különbség volt a két csoport között az alábbi kérdéseknél (mindenik esetben az R-csoport teljesített jobban):

- 5–8 kérdések, az első 4 művelet, $p=0,001<0,05$
- A 17. kérdés, a hasonlítás műveletek száma 5 hosszú számsorozat esetén, $p=0,0001<0,05$
- 25. kérdés, a hasonlítás műveletek száma n elemű számsorozat esetén, $p=0,04<0,05$

Az 1–4 kérdéseknél (a buborékredezésre vonatkozóan) azt vettük észre, hogy az R-csoport 16 tagja kapott 2 pontot, ami azt jelenti, hogy következetesen jól jelölték meg az elempárokat, de rossz műveletet társítottak hozzájuk. Ez azért furcsa, mert ha már beazonosítottuk a kurrens elempárt, akkor a társítandó művelet magától értetendő kellene, hogy legyen: ha megfelelő a két elem sorrendje, akkor „hasonlít”, ha nem, akkor „hasonlít+csere” műveletekre van szükség. Az S-csoport diákjai közül csupán 3 diák vétett ilyen típusú hibát. Egy lehetséges magyarázat erre a különbségre az lehet, hogy a buborékredezés videóban a „hasonlít+csere” művelet párnak megfelelő tánc-

lépésben nincsenek jól elkülöníthető „hasonlít” és „csere” szakaszok (reprezentációs figyelemelterelő). Másfelől, a buborékrendezés animációban a „hasonlít” műveletet képviselő animációk következetesen ugyanazok, függetlenül attól, hogy különálló „hasonlít” műveletről van szó, vagy „hasonlít+csere” műveletpárról.

Érdekes az is, hogy ezt a jelenséget nem tapasztaltuk a kiválasztó rendezés esetében. Ennek oka az lehet, hogy abban az esetben a videóra is igaz az, hogy a „hasonlít” műveletek következetesen jól elkülöníthető tánclépésekkel vannak illusztrálva. Valószínű, hogy ebből az okból kifolyólag is, a kiválasztó rendezés esetében, az első 4 műveleten, az R-csoport szignifikánsan jobban teljesített, mint az S-csoport.

A rendezési algoritmusokra „legjobb eset” -ben („best case”) az jellemző, hogy csak „hasonlít” műveleteket tartalmaznak. A lehetséges ok, amiért az R-csoport szignifikánsan gyengébben teljesített a buborékrendezés idekapcsolódó kérdésein (11. és 19. kérdések) megint csak az lehet, hogy a reprezentációs figyelemelterelők miatt, e csoport tagjai nehezen tudtak elképzelni egy „csere” művelet mentes scénáriót.

Ami a harmadik kutatási kérdést illeti, leszögezhetjük, hogy a tánckoreográfiák esetében, amíg az emberi mozgás jelenléte ellensúlyozhatja a dekoratív figyelemelterelőkből adódó esetleges hátrányt, addig a reprezentációs figyelemelterelők jelenléte gyengítheti ezt a hatást, és negatívan befolyásolhatja a realisztikus ábrázolások hatékonyságát. Ez egy ésszerű következtetés, ugyanis a reprezentációs figyelemelterelők közvetlenül azokat a mozgásokat árnyékolják be, amelyek az algoritmus kulcsműveleteit illusztrálják, más szóval az emberi mozgás dekódolási folyamatára vannak negatív hatással.

3.4.2. Lányok versus fiúk

A fiúk és lányok utóteszt eredményeinek az összehasonlítására kétszempon-
tos varianciaanalízist (ANOVA) használtunk, amelyben a két független válto-
zónak a tanmenet típusát és a nemet, függő változónak pedig az utóteszt telje-
sítményt választottuk. A vizsgálat azt mutatta, hogy egyik algoritmus eseté-
ben sem volt összefüggés a két független változó között, más szóval mindkét
nem egyformán teljesített, függetlenül attól, hogy milyen tanmenet szerint te-
kintette meg az algoritmusokat. Ez az eredmény összecseng korábbi tanulmá-
nyok következtetéseivel (Ahadi et al., 2017), melyek alapján kijelenthető,
hogy nem függ a tanulók nemétől a diákok teljesítménye programozásorien-
tált vagy számítógépesgondolkodás-orientált környezetekben.

3.5. Következtetések

Ebben a fejezetben a sematikus, illetve realisztikus dinamikus vizualizációkat
vettük nagytű alá, egészen pontosan összehasonlítottuk az AlgoRhythmics
környezethez tartozó absztrakt animációkat és realisztikus videókat, mint esz-
közöket a számítógépes algoritmusok oktatásához. Az AlgoRhythmics videók
egyedi sajátossága, hogy tánckoreográfiákkal ábrázolnak keresési és rende-
zési algoritmusokat. Ebből adódik az, hogy hasznosítják az úgynevezett em-
berimozgás-effektust.

A bemutatott kutatás eredményeiből arra következtethetünk, hogy amennyi-
ben egy realisztikus vizualizáció emberi mozgást alkalmaz, ugyanolyan haté-
kony tud lenni, sőt akár hatékonyabb is, mint a megfelelő sematikus ábrázo-
lás. Ez azért fontos következtetés, mert korábbi kutatások inkább a sematikus
vizualizációk hatékonyságát hangsúlyozzák, merthogy kevesebb figyelemel-
terelő elemet tartalmaznak (T1.1 tézis).

Előző eredményekkel összhangban újra megerősítést nyert, hogy a realisztikus ábrázolásokra könnyebb visszaemlékezni és több rokon ábrázolás között könnyebb különbséget tenni. A jelen kutatás kiterjesztette ezt az eredményt az algoritmusvizualizációk területére is (T1.2 tézis).

Definiáltuk, dinamikus vizualizációkra vonatkozóan, a dekorációs és reprezentációs figyelemelterelők fogalmát. Az eredmények arra mutatnak, hogy főleg a reprezentációs figyelemelterelő elemek kerülendők, az a jelenség amikor „árnyék vetődik” azokra a mozgásokra, amelyek az algoritmikai tartalmat illusztrálják (T1.3).

Mindezen következtetések, amelyeket tézisenként is megfogalmaztunk, hasznos információval szolgálhatnak mindazoknak, akik algoritmusvizualizációs környezeteket terveznek és használnak az oktatásban.

4. SEMATIKUS ÉS REALISZTIKUS ALGORITMUSVIZUALIZÁCIÓK ISMÉTELT MEGTEKINTÉSE, ILLETVE EZEK KOMBINÁLÁSA

4.1. Szakirodalmi áttekintés

Ahogy az előző fejezetben már utaltunk rá, egy lehetséges ok, amiért a dinamikus vizualizációkkal való tanulás nem minden esetben hatékonyabb, mint a statikus képek alkalmazása, a videók/animációk tranziens jellegéből adódhat. Ennek egyenes következménye, hogy miközben még zajlik az előző jelenet beépítése egy folyamatosan bővülő képbe (első szakasz), a tanulóknak már azonosítaniuk, és értelmezniük kell az újabb jelenet tartalmazta releváns információkat (második szakasz). Kézenfekvő arra gondolni, hogy elősegítheti a tanulási folyamatot, ha a diákoknak lehetőségük van többször is megtekinteni a vizualizációt. Az ismételt megtekintés hozadéka az lehetne, hogy már nem lenne annyira kritikus a két szakasz egymásra tevődése. Például algoritmusvizualizációk esetén, az első megtekintés alatt, a diákok arra összpontosíthatnak, hogy hogyan vannak a kulcsműveletek illusztrálva, a második megtekintés alatt pedig a cél az lehetne, hogy megértsék az algoritmus stratégiáját, azaz, hogy az egyedi műveletekből összeálljon a globális kép.

Korábbi kutatások azonban arra mutatnak, hogy egyszerűen csak az, hogy valaki kétszer hallgat meg, vagy tekint meg valamit, nem feltétlen eredményez hatékonyabb tanulást. Fritz és társai (2000) megfigyelték, hogy azok az alanyok, akik ugyanazt a szövegrészt kétszer hallgatták meg, egy hét eltolódással, nem feltétlenül tudtak jobban visszaemlékezni a tanultakra, mint azok a társaik, akik csak egyszer hallották. Ezt erősíti meg a Scheiter és társszerzői

(2009) által végzett kutatás is, amely úgy következtet, hogy ugyanannak a vizualizációnak a kétszeri megtekintése nem föltétlenül vezetett jobb megértéshez.

Másfelől azonban, Roy és Chi (2005) azt hangsúlyozzák, hogy bizonyos körülmények között megvalósulhat, hogy egy második megtekintés jelentősen hozzájáruljon a tanulás hatékonyságához. Például, hogy csökkenjen a kétszeri megtekintés miatt fellépő redundancia, egy megoldás az lehetne, hogy ugyanannak az algoritmusnak két különböző, de algoritmikailag izomorf vizualizációját tekintsék meg a diákok (Ainsworth, 2006). Ez elvezet, a többszörös ábrázolással való tanulás témaköréhez, egészen pontosan a többszörös grafikus ábrázolással való tanuláshoz. A többszörös ábrázolással való tanulás leggyakrabban hangsúlyozott előnye az úgynevezett komplementer hatás, azaz, hogy az ábrázolások egymást kiegészítve, egy teljesebb képet nyújtanak a megvizsgált tartalomról.

A korábbi tanulmányok többnyire olyan eseteket vizsgáltak, amikor a tananyag mind auditív, mind vizuális alakban is, vagy mind szöveges, mind képi formátumban is elérhető volt a diákok számára. Ilyen esetekben a többszörös ábrázolás különböző típusú ábrázolások kombinálását jelenti. Mivel ezek feldolgozásáért különböző memóriaterületek felelősek, a túlterhelés veszélye kevésbé áll fenn (Mayer, 2005). Sőt, például, ha szöveges és grafikus ábrázolást alkalmazunk, akkor ennek az az előnye is meglehet, hogy a szöveges rész segíti a grafikus anyag értelmezését. Ettől eltérően, a többszörös grafikus vizualizációk alkalmazásakor, lévén szó ugyanolyan típusú ábrázolásokról, sokkal inkább fennáll a túlterhelés veszélye (Rau, Michaelis & Fay, 2015). A többszörös grafikus ábrázolással való tanulás alapfeltétele, hogy a tanulók világosan megértsék mindkét vizualizációt, és könnyen kapcsolatot tudjanak teremteni közöttük.

Ugyancsak kiemelt figyelmet érdemel az a kérdés, hogy hogyan ütemezzük a vizualizációkat (Rau et al., 2013). Leggyakoribb az egymás utáni megtekintés, de voltak vizsgálatok arra vonatkozóan is, amikor a két vizualizáció párhuzamosan, a képernyő bal és jobb oldalán volt megjelenítve. Egymás utáni megtekintés esetén, ami a sorrendet illeti, előnyös lehet, hogy az egyszerűbb, a kifejezőbb, a megszokottabb ábrázolás előzi meg az összetettebbet, az elvontabbat, az újszerűbbet. Ilyenkor megvalósulhat az, hogy az első ábrázolás hozzájárul a második értelmezéséhez (Ainsworth, 2006).

4.1.1. Sematikus és realisztikus dinamikus vizualizációk kombinálása

A harmadik fejezetben utaltunk két tanulmányra, amelyben a kutatók a többszörös grafikus ábrázolást sematikus és realisztikus dinamikus vizualizációk kombinálása révén valósították meg. Az egyik kutatásban a négy tanmenet a következő volt: „sematikus”, „realisztikus”, „sematikus + realisztikus” (egymást követően) és „sematikus + realisztikus” (párhuzamosan) (Nugteren et al., 2014). A másik kutatásban Scheiter és munkatársai (2009) a következő tanmenetek egymáshoz viszonyított hatékonyságát vizsgálták: „sematikus + sematikus”, „realisztikus + realisztikus”, „sematikus + realisztikus” és „realisztikus + sematikus”. Mindenik esetben a vizualizációk egymást követően voltak megjelenítve.

A jelen kutatásban azt vizsgáltuk, hogy ez utóbbi kísérlet milyen eredményekhez vezetne, ha a tanulás tárgyát számítógépes algoritmusok képeznék (például a Shell-sort algoritmus), és a realisztikus vizualizáció az AlgoRythmics környezet révén történne (13. ábra).



13. ábra: Statikus vs. dinamikus ábrázolásmód

4.2. Kutatási kérdések

A szakirodalmi áttekintés nyomán az alábbi kutatási kérdéseket fogalmaztuk meg:

1. Az emberi mozgás effektusból adódó plusz miatt, hatékonyabbak-e azok a tanmenetek, amelyek magukba foglalnak tánckoreográfia ábrázolásokat?
2. A komplementer hatásból adódó előnyök miatt, jobban teljesítenek-e azok a tanulók, akik kétféle ábrázolást (animációt és tánckoreográfiát) tekintettek meg ugyanarról az algoritmusról, mint azok, akik ugyanazt az ábrázolást látták kétszer (animációt vagy tánckoreográfiát)?
3. Ami a sorrendiséget illeti, előnyösebbek-e azok a tanmenetek, amelyek az egyszerűbb, azaz a sematikus (absztrakt animáció) ábrázolással kezdődnek?
4. Tekintettel arra, hogy a realiztikus ábrázolások könnyebben felidézhetők, jobb eredményeket érnek-e el azok a tanulók, akik másodként a tánckoreográfia ábrázolást látták?

4.3. KÍSÉRLET

Erre a kísérletre a 2018/19-es tanév elején került sor, és három szakaszt foglalt magába: előteszt, tanulási szakasz és utóteszt.

4.3.1. A kutatás résztvevői

Az előtesztnek szánt felmérésre meghívtunk minden elsőéves reálszakos diákot. A felhívásra 151 hallgató válaszolt. A résztvevőket két csoportba osztottuk az alapján, hogy rendelkeznek-e vagy sem alapvető programozói előismeretekkel (középiskolai tanulmányaik nyomán). Azok számára, akik előzőleg nem tanultak informatikát, egy programozási ismereteket nem igénylő számítógépes gondolkodási tesztet állítottunk össze. A jelen kutatásba azt a 84 hallgatót (14% lány) vontuk be, akiknek már volt legalább valamennyi programozói tapasztalatuk (1, 2 vagy 4 évig tanultak informatikát a középiskolában). Számukra az előteszt a kiválasztó és a buborékrendezésekre vonatkozott. Mindkét algoritmus szerepelt az összes résztvevő középiskolás tantervében.

Ahhoz, hogy négy, statisztikailag egyenértékű csoportot alakítsunk ki, rendeztük a résztvevőket az előteszt eredményeik alapján, majd annak függvényében soroltuk őket csoportokba, hogy a sorszámuk 4-gyel való osztási maradéka 0, 1, 2 vagy 3 volt.

4.3.2. A kutatás megvalósításának eszközei

Az előteszthez a résztvevők megkapták mindkét algoritmus pszeudokódját, és tíz-tíz kérdést kellett megválaszolniuk (B függelék: A buborék- és kiválasztó rendezések pszeudókódja). Ezek egyválaszos zárt kérdések voltak, melyek megbízhatóságát a Cronbach-alfa tesztel ellenőriztük ($\alpha=0,85$).

A kutatás a Shell-rendezésre épült, hisz ezt az algoritmust korábban nem tanulta egyik résztvevő sem. A tanulási szakaszban bemutatásra került realiztikus vizualizáció az algoritmus tánckoreográfiája volt. Mivel az Al-goRythmics számítógépes animációk akkor még fejlesztés alatt álltak, sematikus ábrázolásként egy jó minőségű, nagy nézettséggel rendelkező YouTube videót választottunk (New Media, 2013).

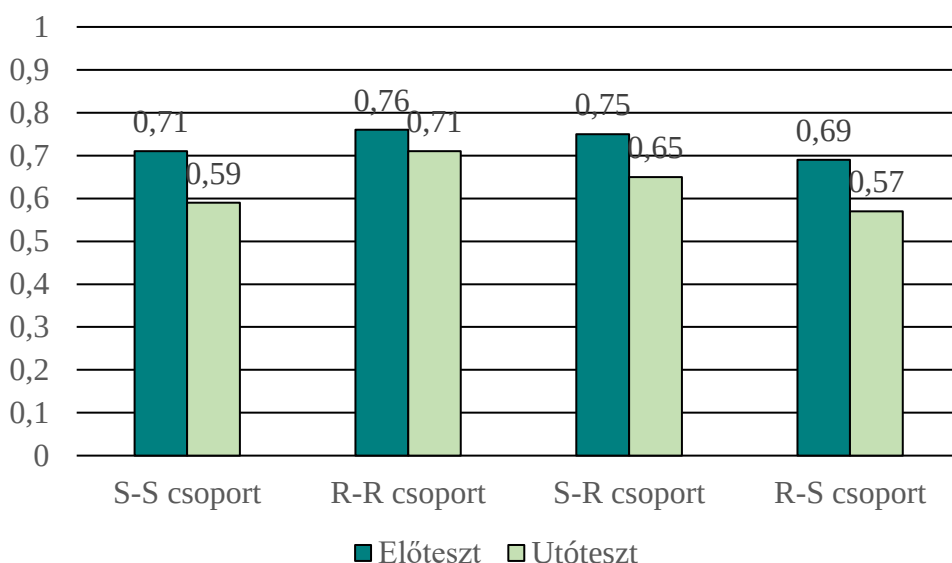
Az utóteszt 14 kérdése azt szándékozott felmérni, hogy a résztvevők mennyire értették meg az algoritmust, azaz el tudják-e képzelni a működését egy újabb véletlen számsorozaton, illetve növekvő („best-case”) és csökkenő („worst-case”) számsorozatokon.

4.4. Eredmények és kiértékelés

A négy csoport előteszt eredményeinek összehasonlításához egyszempontos varianciaanalízist (ANOVA) használtunk. A független változó a tanmenet típusa volt, a függő változó pedig a résztvevők előteszt teljesítménye (3. táblázat: A csoportonkénti eredmények átlagai és zárójelben a megfelelő szórás, 14. ábra: A csoportonkénti eredmények átlagainak ábrázolása). Az elemzés arra mutatott, hogy a négy csoport között nincs szignifikáns eltérés ($F(3,79) = 0,49$), azaz statisztikailag egyenértékűnek tekinthetők.

	S-S csoport	R-R csoport	S-R csoport	R-S csoport
Előteszt	0,71 (0,2)	0,76 (0,21)	0,75 (0,2)	0,69 (0,25)
Utóteszt	0,59 (0,14)	0,71 (0,15)	0,65 (0,2)	0,57 (0,16)

3. táblázat: A csoportonkénti eredmények átlagai és zárójelben a megfelelő szórás értékek



14. ábra: A csoportonkénti eredmények átlagainak ábrázolása

Az utóteszt teljesítmények kiértékeléséhez variancia-kovarianciaelemzést (ANCOVA) használtunk. Ez esetben is a független változó a tanmenet típusa volt, a függő változó viszont az utóteszt teljesítménye. Kovariánsként kezeltük az előteszt pontszámokat. Ez alkalommal a teszteredmények azt mutatták, hogy szignifikáns különbség van a négy csoport között ($F(3,79) = 2,61$, $p = 0,057$, $\eta^2 = 0,09$). Tovább elemezve az eredményeket, arra a következtetésre jutottunk, hogy az utótesztbeli eltérések nem az előtesztbeli minimális különbségek számlájára írhatók ($F(1,79) = 2,04$, $p = 0,15$, $\eta^2 = 0,025$). A négy csoport közötti különbség okainak kiderítése érdekében kontrasztvizsgálatot alkalmaztunk. Az alábbi összehasonlításokat végeztük el:

- S-S csoport versus R-R csoport: szignifikáns különbséget találtunk ($t(79) = 2,24$, $p = 0,027$);
- S-R csoport versus R-S csoport: nem találtunk szignifikáns különbséget ($t(79) = 1,56$);

- S-S és R-R csoportok versus S-R és R-S csoportok: nem találtunk szignifikáns különbséget ($t(79) = 1,11$);
- R-S és R-R csoportok versus S-R és S-S csoportok: nem találtunk szignifikáns különbséget ($t(79) = 0,48$);
- R-S és S-S csoportok versus S-R és R-R csoportok: szignifikáns különbséget találtunk ($t(79) = 2,69$, $p = 0,008$).

Az első kutatási kérdésünkkel kapcsolatban észrevehető, hogy az R-S csoport teljesített a leggyengébben, bár a megfelelő tanmenet magába foglalt tánckoreográfia vizualizációt (emberi mozgást alkalmazó realisztikus ábrázolást). Ez arra enged következtetni, hogy nem jelenthető ki, hogy az emberi mozgás effektust hasznosító ábrázolások, bármely kombinációban jelenjenek is meg, hatékonyabb tanulást eredményeznek. Az először a realisztikus (tánckoreográfia), majd a sematikus (animáció) sorrend szembe ment a multimédiával támogatott oktatás azon alapelveivel, hogy az első vizualizáció kell, hogy legyen az egyszerűbb, a nyilvánvalóbb, a tanulók előzetes ismereteihez jobban igazodó (Ainsworth, 2006). Amíg minden résztvevő látott már különböző absztrakt animációkat, a legtöbbjük számára újdonság számba ment a tánckoreográfiába kódolt algoritmus gondolata.

Másfelől viszont, az eredmények arra mutatnak, hogy azok a diákok, akik kétszer látták a realisztikus ábrázolást, szignifikánsan jobban teljesítettek, mint azok, akik kétszer látták a sematikus. Ez megerősíti azt a következtetésünket, amelyet a harmadik fejezetben bemutatott kutatás alapján vontunk le (T1.1 tézis), hogy az emberi mozgást alkalmazó realisztikus algoritmusvizualizációk (mint amilyenek az AlgoRhythmics videók) hatékonyabbak tudnak lenni, mint a megfelelő sematikus ábrázolások (például absztrakt számítógépes animációk). Másfelől e különbséghez az is hozzájárulhatott, hogy a kétszeri megtekintésből többet nyert az R-R csoport, mint az S-S. Bár ez az

eredmény, amennyiben csak arra összpontosítunk, hogy sematikus versus realisztikus, ellentmond a Scheiter és társai (2009) következtetésének (abban a kutatásban a realisztikus ábrázolással tanuló diákok teljesítettek a leggyengébben), összhangban van azokkal a friss kutatásokkal (ahogy erre az előző fejezetben már utaltunk), amelyek azt hangsúlyozzák, hogy ha egy dinamikus ábrázolás emberi mozgást alkalmaz, akkor jelentősen növekszik a tanulás hatékonysága (Paas & Sweller, 2012).

Ami a második kutatási kérdésünket illeti, nem igazolódott be, hogy a komplexitás elvével összhangban, azok a résztvevők, akik kétféle ábrázolásban is láthatták az algoritmust, jobban teljesítettek volna, mint azok, akik ugyanazt a vizualizációt kétszer tekintették meg. Azt tapasztaltuk, hogy mind az ismételt megtekintésnek, mind több izomorf dinamikus grafikus ábrázolás alkalmazásának megvan az előnye és a hátránya, ezért egyik típusú tanmenet sem jelenthető ki hatékonyabbnak. Ez az eredmény összhangban van a Scheiter és társszerzői idevágó következtetésével, és azt sugallja, hogy mindkét tanmenet típusnak meg lehetnek az előnyei, illetve hátrányai. Ha két különböző vizualizációval tanul a diák, akkor ennek nyilvánvaló előnye, hogy az egyik a másikat kiegészítheti, hátránya viszont az, hogy két különböző vizualizációt kell a diáknak dekódolnia, és megtalálnia a kapcsolatot ezek között. Ismételt megtekintés esetén a második lejátszás segítheti a diákot abban, hogy megértse azt, amit az első lejátszáskor elszalasztott. De e tanmenet hátránya az lehet, hogy a második megtekintés az első árnyékába kerülhet (a diákok úgy viszonyulhatnak hozzá, hogy „áh, ezt már láttam egyszer”), és összpontosított figyelés hiányában elmarad az összkép kialakítása.

A harmadik kutatási kérdés tekintetében az eredmények arra mutatnak, hogy az a csoport, amely először nézte meg a sematikus ábrázolást, majd utána a realisztikust, valóban jobban teljesített, mint azok a résztvevők, akik először a realisztikus majd a sematikus vizualizációt látták. Azonban a különbséget

nem találtuk szignifikánsnak. Ez megint csak összhangban van a Scheiter és társszerzői (2009) végezte kutatással, akik ugyanerre a következtetésre jutottak. Mindez arra enged következtetni, hogy valamilyen szinten jelen lehetett az a jelenség, hogy a sematikus vizualizáció hozzájárult ahhoz, hogy a realisztikus ábrázolást könnyebben megértésük a diákok, de ez a hatás nem volt annyira erős, hogy teljesen felülkerekedjen azon a plusz leterheltségen, ami egy újabb ábrázolás megértésével, és a kettő összekapcsolásával együtt járt.

A negyedik kutatási kérdés kapcsán szignifikáns eredményhez jutottunk.

Azok a csoportok, amelyek második vizualizációként a realisztikus ábrázolást tekintették meg (R-R és S-R), szignifikánsan jobban teljesítettek, mint azok, akik a sematikus ábrázolást látták másodikként (S-S és R-S). Egy lehetséges magyarázat erre az lehet (és ez újra összhangban van előbbi kutatásokkal, illetve a T1.2 tézissel), hogy a realisztikus ábrázolások könnyebben megjegyezhetők, és ha a tanulás a realisztikussal fejeződik be, akkor könnyebb visszaemlékezni az algoritmus stratégiájára.

4.5. Következtetések

Ebben a kutatásban azt vizsgáltuk, hogy amennyiben ugyanarról az algoritusról két ábrázolással is rendelkezünk, sematikussal és realisztikussal is, akkor ezek mely kombinációja eredményez hatékonyabb tanulást. Az eredmények arra mutatnak, hogy realisztikus ábrázolások esetén kiváltképpen hasznos lehet egy második megtekintés. Az a tény, hogy azok a diákok, akik kétszer tekintették meg a realisztikus ábrázolást szignifikánsan jobban teljesítettek, mint azok, akik a sematikus látták kétszer, nem csak arra enged következtetni, hogy az emberi mozgást alkalmazó realisztikus ábrázolások hatékonyabb eszközök, mint a megfelelő sematikusak, hanem arra is, hogy a kétszeri megtekintésnek több hozadéka volt az R-R csoportnál, mint az S-S csoportnál (T2.1 tézis).

A jelen vizsgálat nyomán, algoritmusvizualizációs környezetben is beigazodott, hogy annak is meglehet az előnye (illetve hátránya), ha ugyanannak az algoritmusnak két izomorf vizualizációját alkalmazzuk (S-R és R-S csoportok), és annak is, ha ugyanazt a vizualizációt kétszer tekintik meg a diákok (S-S és R-R csoportok).

Bár a különbség nem volt szignifikáns (S-R versus R-S csoport), a mi eredményeink is azt sugallják, hogy amennyiben kétféle ábrázolást is alkalmazunk, előnyös lehet először a sematikus (az egyszerűbbet, amelyik jobban igazodik a diákok előzetes ismereteikhez) lejátszani, és azután a realisztikus (az összetettebbet, az újszerűbbet). Újra igazolást nyert, hogy a sajátos jegyeket tartalmazó realisztikus ábrázolásokra könnyebb visszaemlékezni, és ez lehet az oka annak, hogy azok a csoportok, amelyek másodikként a realisztikus vizualizációt látták (R-R és S-R), szignifikánsan jobban teljesítettek, mint a másik két csoport (S-S és R-S) (T2.2 tézis).

5. AZ INTERAKTIVITÁS SZEREPE ONLINE ALGORITMUSVIZUALIZÁCIÓS KÖRNYEZETEK BEN

5.1. Szakirodalmi áttekintés

A dinamikus vizualizációk hatékonyságát vizsgáló szakirodalom egy szeletje az interaktivitás oldaláról vizsgálja e témakört. Egyik szempont, amely több kutatás tárgyát is képezte az, hogy milyen szintű interaktivitás járul leginkább hozzá az animációkkal való tanulás hatékonyságához. A Byrne és társszerzői (1996) nevéhez fűződő metaanalízis úgy következtet, hogy vegyes eredmények születtek e kutatások nyomán. Egyes tanulmányok olyan következtetésre jutottak, hogy amennyiben a diákok be vannak vonva a vizualizációs folyamatba, ez elősegíti a hatékonyabb tanulást. Többségben voltak azonban azok a vizsgálatok, amelyek inkább ennek az ellenkezőjét bizonyították. Egy lehetséges magyarázat, amiért az interaktivitás nem mindig vezet jobb eredményhez az lehet, hogy az animációs folyamat megszakítása akadályozhatja a diákokat a vizualizációban való elmélyülésben. Azok a tanulmányok, amelyek az interaktivitás hatékonyságát hangsúlyozzák, azt emelik ki, hogy ez hozzájárul ahhoz, hogy a diákok tudatosabban kövessék a vizualizációt.

Néhány kutatás kifejezetten az algoritmusvizualizációs környezetekre összpontosított, és kijelenthetjük, hogy e szűkített keretben is felemás eredmények születtek. Az egyik módja, ahogy interaktívvá lehet tenni egy algoritmusvizualizációs folyamatot, ha bizonyos kulcsmomentumokban az animáció megáll, és a diákoknak meg kell mondaniuk, hogy melyik lenne az algoritmus következő lépése. Byrne és munkatársai (1996) implementálta kísérletben valahányszor megállt a videó, a diákoknak szóban kellett megmondaniuk, hogy melyik lenne a következő művelet. E szerzők arról számolnak be,

hogy ezzel a fajta interaktivitással az animációs folyamat hatékonyabb tanulást eredményezett.

Egy hasonló kutatásban (Korhonen & Malmi, 2000) a diákoknak manuálisan kellett rekonstruálniuk az algoritmust, és adott időpillanatokban, egy grafikus editor segítségével, válaszolniuk kellett a kapcsolódó adatszerkezet kurrens állapotára vonatkozó kérdésekre. E kutatás eredményei is arra mutattak, hogy az interaktivitásnak pozitív hozadéka volt a tanulási folyamatra nézve. Egy harmadik kutatás ebben a témakörben Naps és munkatársai (2000) nevéhez kapcsolódik, akik az algoritmusvizualizációs környezetükbe úgynevezett „Stop and think” kérdéseket iktattak be. Ezek a szerzők is úgy következtetnek, hogy abból adódóan, hogy a diákok elgondolkodtak a kérdéseken, és válaszaikra azonnali visszajelzést kaptak, a tanulás eredményessége nagymértékben nőtt.

Másfelől azonban, Jarc és Heller (2000) kutatók éppen az ellenkezőjét tapasztalták az előbbieken tárgyalt jelenségnek. Az ő kísérletükben azok a diákok, akik zavartalanul el tudtak mélyülni az animációban, jobban teljesítettek, mint azok a társaik, akik mindegyre „interaktív előrejelzéseket” („interactive prediction”) kellett, hogy adjanak a következő műveletről.

Az AlgoRythmics környezetben végzett korábbi kutatások is az interaktivitás értékét hangsúlyozták. Példaként megemlíthetjük azt a kutatást, amelyben a szerző humán, reál beállítottságú egyetemi hallgatók teljesítményét vizsgálta (Kátai, 2015). A kísérlet gerincét képező tanmenetben a résztvevők először tánckoreográfiával illusztrálva, majd pedig animálva tekinthették meg a kiválasztott algoritmust. Ezt követően a diákokat felkérték, hogy interaktív módon, úgymond vezényeljék le a teljes algoritmust (összes lépését megadva), először ugyanazon az inputon, amelyen az animáció is zajlott, másodszor vé-

letlen inputon, úgy, hogy látszottak a számok („white üzemmód”), harmadszorra pedig ugyancsak véletlen bemeneten, de úgy, hogy a számok el voltak rejtve („black üzemmód”). Az eredmények azt mutatták (főleg a humán orientált diákok esetében), hogy az interaktív részvétel a vizualizációs folyamatban, jelentősen hozzájárult az algoritmus stratégiájának jobb megértéséhez.

Az előbbieken vázoltak értelmében, az algoritmusvizualizációs környezetek osztályozhatók annak alapján, hogy milyen szintű interaktivitást tesznek elérhetővé. Nevezzük „0-interaktivitásnak” azt, amikor a tanuló egyszerűen csak megtekinti a vizualizációt. Azt a változatot, amikor a tanulóktól azt várják el, hogy csak bizonyos kulcsmomentumokban kapcsolódjanak be a vizualizációs folyamatba, tekintjük „ $\frac{1}{2}$ -interaktivitásnak” („fél interaktivitás”). Legyen „1-interaktivitás” (vagy „teljes interaktivitás”) a neve annak az esetnek, amikor a diákoknak a teljes animációs folyamatot le kell vezényelniük.

A vegyes eredmények, ami a különböző interaktivitási szintek hatékonyságát illeti, önmagukban is azt sugallják, hogy mindenik interaktivitás típusnak meg lehetnek az előnyei, illetve hátrányai. Például a „0-interaktivitás” előnye az lehet, ahogy már utaltunk is rá, hogy a tanulók, úgymond zavartalanul el tudnak mélyülni az animációs folyamatban. Hátrány lehet viszont az, hogy könnyen el is kalandozhatnak a vizualizáció alatt. Ami a „ $\frac{1}{2}$ -interaktivitást” illeti, az hogy a tanulóktól elvárják, hogy előre nem ismert időpillanatokban bekapcsolódjanak a vizualizációba, hozzájárulhat a feszültebb figyeléshez. Amennyiben azonnali visszajelzés- is érkezik, ez elejét veheti annak, hogy a tanulók félreértelmezzenek dolgokat. A visszajelzések, egyben korrekciót is jelentenek. Másfelől viszont az is megtörténhet, hogy ilyen környezetekben a tanulók úgy tekintenek a közbeékelte kérdésekre, mint találgató játékokra, és a remélt, tudatosabb tanulás elmarad. Nyilván az „1-interaktivitás” követeli meg leginkább a tanulóktól a folyamatos, összpontosított figyelmet a tanulási

folyamat alatt. E „teljes interaktivitás” hátránya viszont az lehet, hogy nagyon felaprózódik a vizualizációs folyamat, és nehezebben áll össze a diákok elméjében az összkép, az algoritmus stratégiája.

5.2. Kutatási kérdések

Az előző fejezetben már kifejtettük, hogy haszna lehet egy vizualizáció kétszeri megtekintésnek. Ebben a kutatásunkban azt vizsgáltuk, hogy milyen előnyökkel, illetve hátrányokkal járhat, ha a második megtekintés alatt 0, $\frac{1}{2}$, illetve 1-interaktivitást alkalmazunk. Az alapvető kutatási kérdés, amelyet megfogalmaztunk a következő.

- Létezik-e optimális interaktivitási szint algoritmusvizualizációs környezetekben?

5.3. KÍSÉRLET

Ez a kutatás egyfajta folytatása az előző fejezetben bemutatott tanulmánynak. Arra a 2018/19-es tanév elején, erre pedig a 2019/20-as tanév elején került sor. A 2018/19-es tanév elsőévesein azt mértük, hogy előnyös-e, ha a diákok kétszer is megtekintik egy algoritmus két különböző vagy ugyanazon vizualizációját. Az előteszt számítógépes gondolkodást mérő feladatokon alapult, a tanulási szakasz és az utóteszt pedig a Shell-rendezés köré épült. A most bemutatásra kerülő kísérlet hasonló felépítésű volt, csak hogy a résztvevők a Shell-rendezés AlgoRythmics animációját tekintették meg kétszer (időközben elkészült; az előző kísérletben egy YouTube animációt használtunk sematikus ábrázolásként), és a három csoport, amelyet kialakítottunk, a második megtekintés alatt bekapcsolódott a vizualizációs folyamatba, a 0-, $\frac{1}{2}$ -, illetve 1-interaktivitási szinteknek megfelelően.

5.3.1. A kutatás résztvevői

A kísérletben 134 elsőéves, reálszakos hallgató vett részt, akiknek a 13,97%-a volt lány. A résztvevők véletlenszerűen lettek három csoportba osztva, amelyek egyenlő szintűeknek bizonyultak mind a középiskolában programozással töltött évek átlagai alapján, mind az elő-teszt eredmények tekintetében (4. táblázat: Résztvevők)

	Létszám	Lányok aránya (%-ban)	Középiskolában tanult programo- zást (évek átlaga)	Elő-teszt teljesít- mény (%-ban)
A csoport	46	10,86	2,13	68,13
B csoport	44	15,55	2,20	68,28
C csoport	44	13,33	2,16	67,75

4. táblázat: Résztvevők átlagai csoportokra bontva

5.3.2. A kutatás megvalósításának eszközei

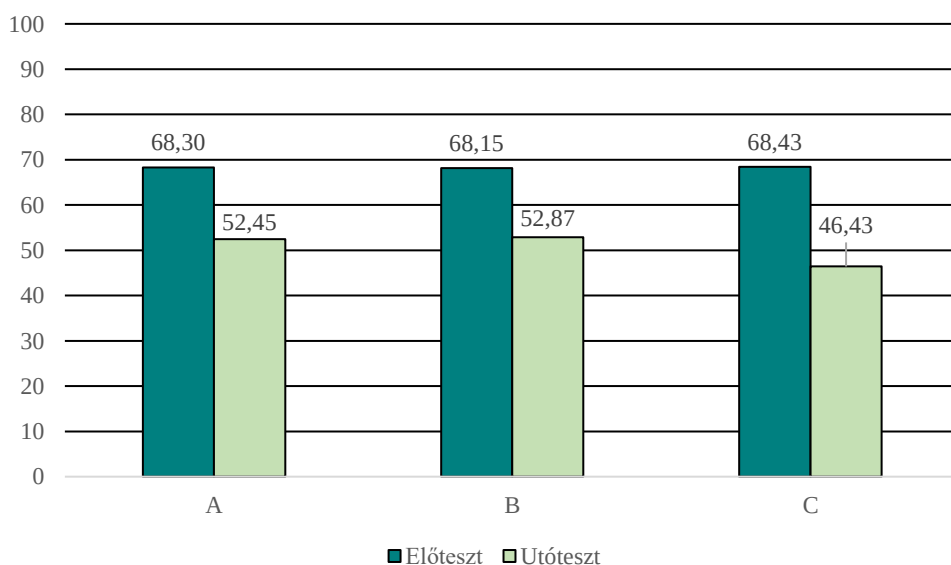
Az előteszt kérdései a Bebras verseny magyarországi weboldaláról származó feladatokon alapultak (A függelék: Számítógépes gondolkodás teszt). A tanulási szakaszhoz, amint már említettük, a Shell-rendezést használtuk fel. A megújult AlgoRythmics környezetben az algoritmusok számítógépes animációi éppen a fentebb definiált három üzemmódban (0-, $\frac{1}{2}$, illetve 1-interaktivitással) tekinthetők meg. Annak mérésére, hogy mennyire értették meg a résztvevők az algoritmus stratégiáját, egy hasonló utóteszt kérdéssort alkalmaztunk, mint az előző fejezetben bemutatott vizsgálat esetében.

5.4. Eredmények és kiértékelés

Az előteszt válaszokat egyszempontos varianciaanalízissel (ANOVA) elemeztük. A független változó az alkalmazott interaktivitási szint volt (nulla,

fél, teljes), a függő pedig a résztvevők pontszáma. A három csoport statisztikailag egyenértékű volt ($p > 0,05$).

Az utóteszt eredmények vizsgálatánál variancia-kovarianciaelemzést (ANCOVA) használtunk. A független változó ugyancsak az alkalmazott interaktivitási szint volt, a függő változó viszont az utóteszt teljesítmény. Kovariánsként kezeltük az előteszt pontszámokat. Ez esetben sem találtunk szignifikáns különbséget a három csoport között. Ily módon a kutatási kérdésünkkel kapcsolatban úgy következtethetünk, hogy nem eredményez számottevő különbséget a tanulási teljesítményben, ha nulla-, fél- vagy teljes-interaktivitást alkalmazunk. Ez összhangban van a korábbi kutatásokkal, abban az értelemben, hogy kimutatták, hogy mindhárom tanmenet típusnak megvannak az előnyei, illetve hátrányai.

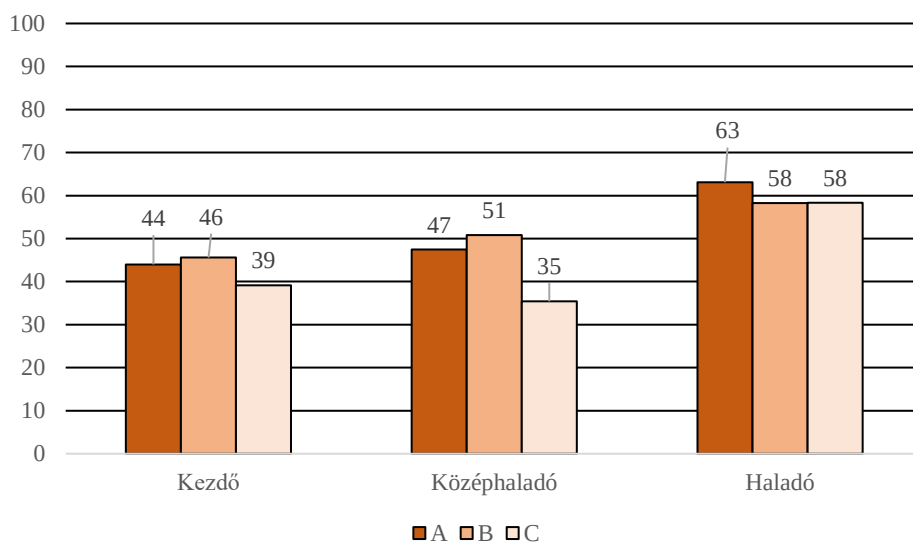


15. ábra: Előteszt és utóteszt eredményei csoportonként

Mivel az előtesztelés során összességében nem találtunk szignifikáns különbséget a különböző csoportok között (15. ábra), megvizsgáltuk a tanulók utóteszten elért teljesítményét az előzetes programozási tapasztalatok függvényében is.

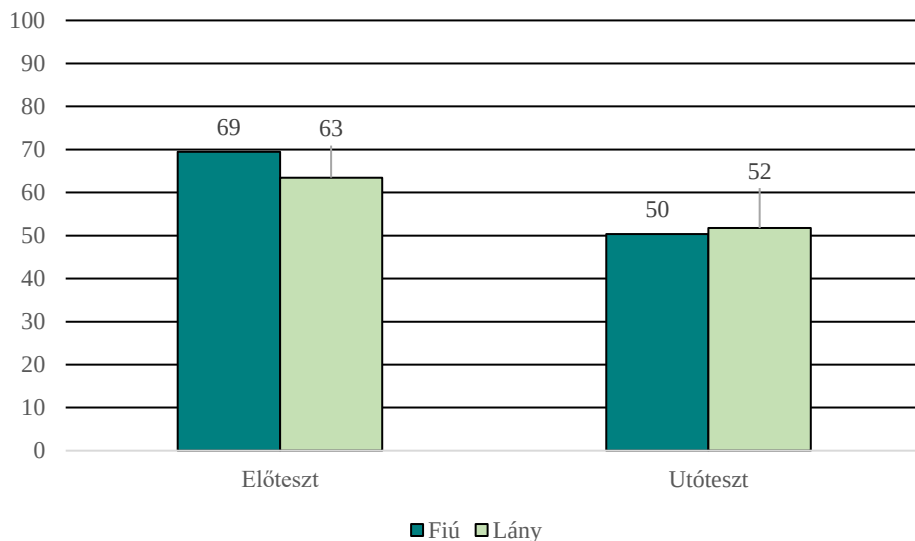
Azoknál a tanulóknál, akik nem rendelkeztek előzetes programozási ismeretekkel (0 évet) vagy csak nagyon keveset (1-2 évet) tanultak programozást, az eredmények lineáris növekedését tapasztaltuk az interaktivitás növekedésének függvényében (44%, 47%, 64% - 0 év programozás; 46%, 51%, 58% - 1 vagy 2 év programozás) (T3.1 tézis).

A haladók (3 vagy 4 év tanultak programozást) eredményeit nem befolyásolta a tanulás interaktivitási szintje (T3.1 tézis) (16. ábra).

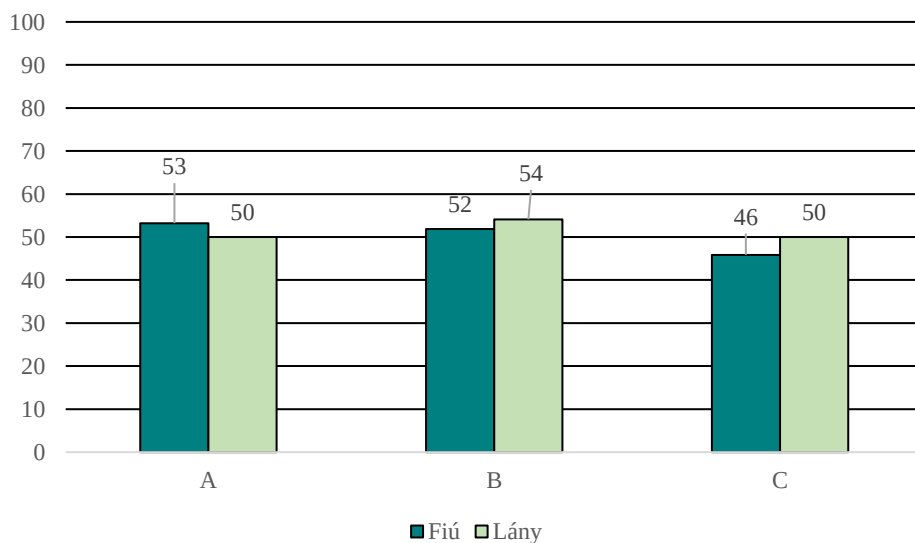


16. ábra: A 3 programozási szinten levő diákok eredményei az utóteszt során, csoportonként

Annak ellenére, hogy az előtesztelés során a fiúk valamivel jobban teljesítettek, mint a lányok, az utótesztelésen, azoknál a csoportoknál, ahol a tanítás interaktívan zajlott, a lányok eredményei a fiúk eredményei a fölé kerekedtek (T3.2 tézis) (17. ábra, 18. ábra).



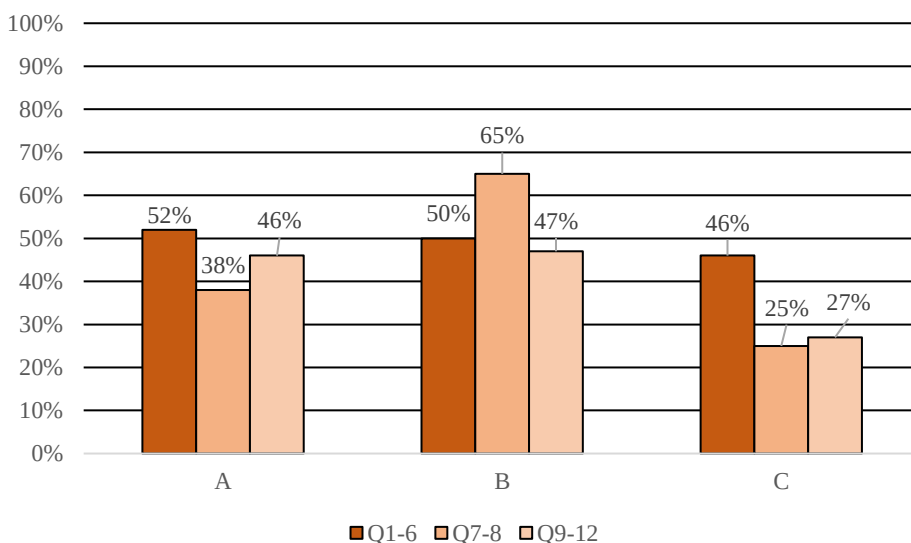
17. ábra: Az előteszt és utóteszt eredményei összesítve nemenként



18. ábra: Az előteszt és utóteszt eredményei nemenként csoportokra lebontva

Az 1 vagy 2 év programozási ismerettel rendelkező tanulók válaszait a kérdések típusa szerinti felbontásban is megvizsgálva, a következő eredményekhez jutottunk:

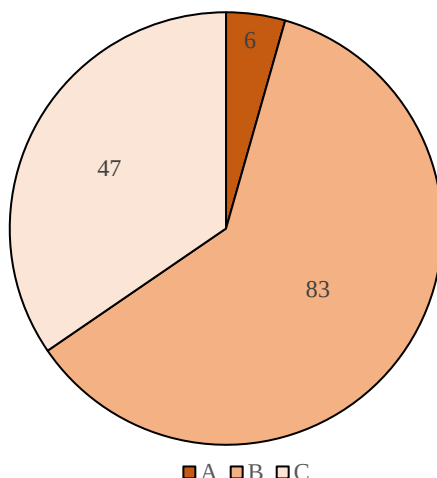
- A 7–8. kérdések az algoritmus egy-egy adott pontján történő művelethez kapcsolódtak. Itt azok a tanulók értek el szignifikánsan jó eredményeket, akik „1/2 interaktivitással” tanultak, vagyis előre nem ismert időpillanatokban kellett bekapcsolódjanak a vizualizációba és ez meghozta gyümölcsét. Ebben az esetben azok a tanulók teljesítettek a legrosszabbul, akik teljes, „1 interaktivitással” tanultak.
- A 9–12. kérdéseknél, amelyek szintéziskérdések voltak, azok a tanulók, akik úgymond zavartalanul el tudnak mélyülni az animációs folyamatban („0 interaktivitás”) valamint azok, akik csak bizonyos kulcsmomentumokban voltak felkérve, hogy kapcsolódjanak be a vizualizációs folyamatba a tanulás alatt („1/2 interaktivitás”) szignifikánsan jobb eredményeket értek el a teljes, „1 interaktivitással” tanuló diákokkal szemben (T3.3 tézis) (19. ábra).



19. ábra: Az 1-6., 7-8., 9-12. kérdések válaszai az 1 vagy 2 évet tanultak esetén

Arra a kérdésre, hogy „Melyik módszer (interaktivitási szint) tűnik számodra a leghatékonyabbnak (függetlenül attól, hogy melyik Tanfolyamcsoportba

tartoztál)?”, a diákok szignifikánsan a $\frac{1}{2}$ interaktivitással való tanulást választották (20. ábra).



20. ábra: A diákok véleménye a leghatékonyabb tanulási módszerről

5.5. Következtetések

Ebben a kutatásban azt tanulmányoztuk, hogy hogyan befolyásolják különböző interaktivitási szintek az online környezetben való tanulás hatékonyságát. Az eredmények arra mutatnak, hogy a 0, vagy csupán 1 vagy 2 év előzetes programozási ismerettel rendelkező diákok javára válhat az algoritmusvizualizáció interaktív bemutatása, a haladók (3 vagy 4 év programozás) azonban otthonosan mozognak bármely interaktivitási szinttel rendelkező tanfolyam esetén (T3.1 tézis).

Nincs szignifikáns különbség a különböző interaktivitási szintekkel tanuló fiúk és lányok eredményei között. Mivel a lányok eredményei, nem szignifikánsan, magasabb értékeket mutattak a felmérés során, ezért számukra előnyt jelenthet, ha interaktívan tanulhatnak (T3.2 tézis).

A teljes interaktivitással tanuló diákoknak nem sikerült összeállítani egy összképet a tanult algoritmusról, így a szintézis kérdések megválaszolásában is lemaradtak (T3.3 tézis).

6. TANÁRI KÉRDÉSSOROK ALGORITMUSVIZUALIZÁCIÓS KÖRNYEZETEKBE

6.1. Szakirodalmi áttekintés

A saját tapasztalataink és kollegáink visszajelzései is azt hangsúlyozták, hogy a diákok, amennyibe magukban használják az AlgoRhythmics környezetet, csak részlegesen tudják kiaknázni a potenciált, amit magában rejt. Például azok a definíciók, amelyek próbálják körül írni a számítógépes gondolkodás fogalmát, hangsúlyozzák, hogy fontos, hogy a tanulók a tanulmányozott algoritmusok hatékonyság-vizsgálatában is elmélyüljenek (Shute et al., 2017).

Egy diák, aki úgymond autodidakta módon használja a környezetet, talán megérti a vizsgált algoritmus stratégiáját, de nem valószínű, hogy egyáltalán gondol is hatékonysági szempontokra. Viszont egy vegyes oktatási környezetben („blended learning”), a tanár rá tudja vezetni a tanulókat e szempontokra (Graesser & Person, 1994). Persze fontos, hogy ne sérüljön az aktív tanulás elve, és a tanár megmaradjon irányítói szerepkörében azáltal, hogy támogatja a diákokat, hogy egy adott irányba gondolkozzon, és olyan tartalmakat is érzékeljen, amire magától nem igazán jönne rá (Guzdial & Tew, 2006).

Egy hatékony módja annak, hogy a tanár úgy serkentsen a diákokat, hogy közben megmaradjon az aktív tanulás élménye az lehet, ha a tanár ügyesen kérdez. Régtől fogva elismert a kérdések értéke az oktatásban. Természetesen a kérdéseknek, hogy valóban elérjék céljukat, jól átgondoltnak kell lenniük, és az sem mindegy, hogy milyen sorrendben tesszük fel ezeket (Tofade et al., 2013). Fontos, hogy a kérdések progresszív módon épüljenek egymásra. Ha

olyasmiből indulunk ki, ami igazodik a tanulók jelenlegi ismereteihez, és lépésről lépésre vezetjük őket az új ismeretek fele, akkor sikerülhet a felfedezés élményét nyújtani nekik.

A 4. fejezetben érintettük a többféle ábrázolással való tanulás témakörét.

Utaltunk rá, hogy milyen előny származik abból, ha a grafikus ábrázolás mellé szöveges tartalom is társul: ez utóbbi megvezetheti a tanulót az előbbi értelmezésében. Gyakran a dinamikus grafikus ábrázolásokhoz társított narrációnak is ez a szerepe (Berney & Bétrancourt, 2016). Amikor kérdéssorozatot társítunk egy vizualizációhoz valami hasonló valósul meg: a kérdések segíthetnek a grafikus ábrázolás teljesebb megértésében.

Amint fentebb utaltunk rá, a kérdések előnye, hogy a tanuló megmarad a tanulási folyamat főszereplőjének. Megszokott, hogy a kérdés-felelet megbeszélés a tanár felügyelete, a közvetlen irányítása alatt zajlik. Még inkább háttérben marad a tanár, ha egy jól átgondolt kérdéssorozatot társít a vizualizációhoz, amit a diákok magukban kell, hogy megválaszoljanak. Ahhoz, hogy elég legyen egy ilyen szintű tanári beavatkozás, a vizualizációnak kellően ki-fejezőnek kell lennie.

Az előző fejezetekben bemutatott kutatások mindenikében megjelentek a vizualizációhoz társított, algoritmus bonyolultsági szempontokra is kitérő kérdéssorozatok. Ezekre minden esetben úgy tekintettünk, mint utóteszt kérdésekre, amelyek a vizualizációval való tanulás hatékonyságát mérték le. Ebben a kutatásban, amely időrendben megelőzte az előbbieket, a szóban forgó kérdéssorozatok egyben a tanulási folyamat részének is tekintjük. E kutatás egy másik sajátossága, hogy a tanmenet kiegészült tanár irányította kérdés-felelet megbeszéléssel is.

6.2. Kutatási kérdések

E kutatás magvát képező kísérletet háromszakaszosra terveztük. Az első szakaszban a résztvevők megtekintették három algoritmus AlgoRythmics vizualizációját, a másodikban egy csoportszintű, osztálytermi beszélgetésre került sor, a harmadik szakaszban pedig egy újabb algoritmusvizualizációval szembesültek a diákok. A három szakaszban más-más módon jelentek meg a kérdések. Az első és harmadik szakaszokban, a tanulók magukban mélyülhettek el a vizualizációkhoz csatolt kérdéssorozatokban. A második szakasz keretében a kérdéssorozatot a tanár tette fel.

Konkrétabban, az első és harmadik szakaszok kérdéssorozatai azzal a céllal lettek összeállítva, hogy rávezessék a diákokat algoritmus-bonyolultsági szempontokra. Ezzel összhangban a résztvevők már a vizualizációk megtekintése előtt megkapták a kérdéseket. Ami a második szakasz kérdéssorozatát illeti, ennek az volt a célja, hogy a diákok úgymond rádöbbenjenek az első szakaszbeli kérdések helyes válaszára, illetve ezek miértjeire.

Az alábbi kutatási kérdéseket fogalmaztuk meg:

1. Elég kifejezőek-e az AlgoRythmics vizualizációk ahhoz, hogy egy gondosan összeállított kérdéssorozat rá tudjon vezetni programozói előismeretekkel nem rendelkező elsőéves hallgatókat algoritmus-bonyolultsági tartalmakra?
2. Hozza tud-e járulni a második szakaszbeli kérdés-feleletes megbeszélés ahhoz, hogy a harmadik szakaszban jobban teljesítsenek a résztvevők, mint ahogy az első szakaszban teljesítettek?

6.3. KÍSÉRLET

E kutatásra a 2017/18-as tanév elején került sor.

6.3.1. A kutatás résztvevői

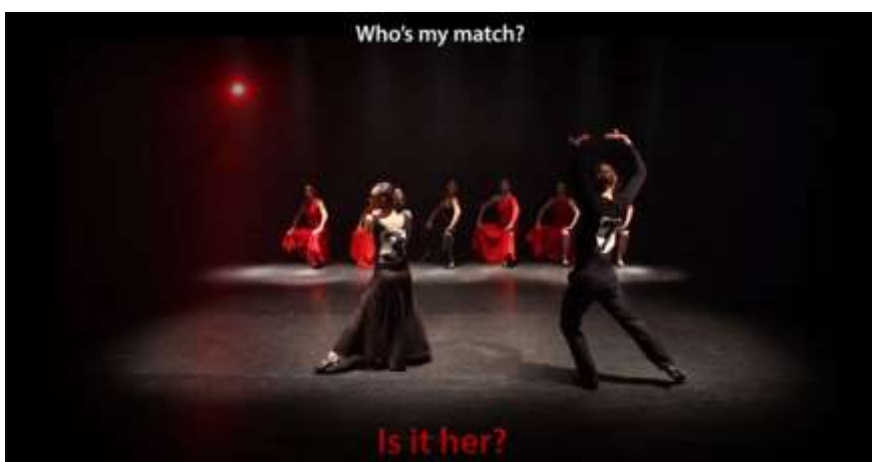
A kísérletben, ez esetben is, minden elsőéves, reálszakon tanuló hallgatót bevontunk (181 diák, 13% lány). A résztvevőket annak alapján soroltuk csoportokba, hogy hány évig tanultak programozást a középiskolai tanulmányaik alatt: A-csoport (0 évet tanultak, 27%); B-csoport (1 vagy 2 évet tanultak, 25%); C-csoport (4 évet tanultak, 48%). Az B-csoport esetében a középiskolai tanterv kitért a két alapvető keresési algoritmusra (lineáris és bináris), illetve tartalmazta néhány négyzetes időbonyolultságú rendezési algoritmus vizsgálatát, alapszinten. Ily módon, ezek a résztvevők, bár ismerhették a szóban forgó algoritmusok stratégiáját, az algoritmus-bonyolultság fogalma új volt számukra. Azon csoport tagjai, akik 4 évet tanultak (C-csoport), az algoritmusok stratégiája mellett, bonyolultsági fogalmakkal is megismerkedtek a középiskolában.

A kísérletünk első szakaszához az A-csoportot tekintettük kísérleti csoportnak, a B-t pedig kontroll csoportnak. Kíváncsiak voltunk rá, hogy megvan-e az AlgoRhythms környezetben a potenciál ahhoz, hogy az A-csoport diákjait felzárkóztassa a B-csoportéhoz. A C-csoport, ebben az elemzésben, egyfajta másodlagos kontroll csoportként lett bevonva.

A kísérlet második és harmadik szakaszában csak az A-csoport tagjai vettek részt. Arra számítottunk, hogy miután e résztvevők elmélyültek az első szakasz kérdéseiben, és aktívan részt vettek a második szakasz keretében zajló megbeszélésen, a harmadik szakaszban szignifikánsan jobban fognak teljesíteni, mint ahogy az első szakaszban tették (e két szakasz megfelelő kérdései kapcsán).

6.3.2. A kutatás megvalósításának eszközei

Az első szakasz a lineáris és bináris keresések, valamint a buborékrendezés vizualizációira épült (7, illetve 10 elemű véletlen számsorozatokon illusztrálva az algoritmusok működését). Mivel a keresési algoritmusok viszonylag egyszerűek, ezek csak az AlgoRhythmic ténckoreográfiákkal voltak illusztrálva (21. ábra: Lineáris keresés flamenco tánccal, 22. ábra: Bináris keresés flamenco tánccal).



21. ábra: Lineáris keresés flamenco tánccal



22. ábra: Bináris keresés flamenco tánccal

A résztvevők ezeket kétszer tekinthették meg. A buborékrende­zés esetében először a tánckoreográfiát, majd egy számítógépes animációt játszottunk le. Mindhárom algoritmushoz igényesen megfogalmazott, gondosan sorba sze­dett kérdéssorozat társult, amelyek az algoritmusok legjobb („best case”) és legrosszabb („worst case”) esetekben való viselkedésére vonatkoztak. Az igé­nyes megfogalmazás alatt olyasmit értünk, hogy például, a „legjobb” és „leg­rosszabb” szavakat lecseréltük azzal, hogy „legszerencsésebb”, illetve „leg­szerencsétlenebb”, vagy a „hasonlítás művelet” kifejezés helyett sok esetben az „összehasonlási jelenet” kifejezést használtuk.

A kísérlet sajátossága az volt, hogy a résztvevők már a vizualizációk előtt megkapták a kérdés-sorozatokat. A kétszeri megtekintéssel az volt a célunk, hogy az első megtekintés alatt a résztvevők hozzászokhassanak a környezet­hez, és hogy felfoghassák, hogy miként vannak eltáncolva, illetve animálva az algoritmusok kulcsműveletei (hasonlít és csere). Kifejezetten ösztönöztük őket arra, hogy csak a második megtekintés alatt osszpontosítsanak az algo­ritmus stratégiájára és a megválaszolandó kérdésekre. Az első szakasz egy szintéziskérdéssel zárult, amely a megvizsgált algoritmusok egymáshoz vi­szonyított hatékonyságára vonatkozott.

A második szakaszban a tanár olyan kérdéseket tett fel, mint például:

- Mi a legjobb/legrosszabb-eset scenárió a lineáris/bináris keresés ese­tében akkor, ha a keresett elem garantáltan részét képezi a sorozatnak, hát akkor, ha hiányozhat is?
- Ha a számsorozatot, amelyben a keresés zajlik, úgy fogjuk fel, mint keresési teret, akkor minden egyes összehasonlítás (amely úgy dől el, hogy nem a keresett elemről van szó), hány elemet zár ki, a lineá­ris/bináris keresési stratégiák esetében, a keresési térből.

- Miután beazonosítottuk a bináris kereséshez tartozó bináris fát, a tanulók rá voltak vezetve a következő képletre: $2^0 + 2^1 + \dots + 2^{k-1} = n$; $k = \log_2(n+1)$.
- A buborékrendezés algoritmus legrosszabb esetben viselkedése kapcsán a következő képlet lett elemezve: $(n-1) + (n-2) + \dots + 2 + 1 = n(n-1)/2$.

A harmadik szakaszban csak az A-csoport tagjai vettek részt, amely egy újabb algoritmusra épült, a kiválasztó rendezésre (tánc-koreográfiával és számítógépes animációval vizualizálva). A tanmenet ugyanaz volt, mint az első szakaszbeli buborékrendezés esetében.

6.4. Eredmények és kiértékelés

A bemutatott ismeretanyagot, mindenik algoritmus kapcsán, két részre bontottunk. A diákok meg kellett, hogy értsék, az algoritmus stratégiáját vagy logikáját (I. típusú ismeret), majd erre az ismeretre építve, el kellett képzelniük az algoritmusoknak a legjobb, illetve a legrosszabb esetekre vonatkozó viselkedését (II. típusú ismeret). Az I. típusú ismeretet a diákok hangsúlyosan az algoritmusvizualizációból kellett kinyerjék, a II. típusú ismeret megszerzésében viszont sokat segít nekik a csatolt kérdéssorozat. Az A-csoport tagjainak nem volt semmilyen előzetes ismerete, a B-csoportnak volt I. típusú előzetes ismerete, a C-csoport tagjainak úgy I., mint II. típusú ismerete is volt (középiskolai tanulmányaikból adódóan). Bár a kérdésekre adott válaszok a II. típusú ismeretet tükrözték, ez I. típusú ismeretre épült.

Az adatok kiértékelésénél figyelembe vettük mindhárom csoportnak az első szakaszban adott válaszait, illetve összehasonlítottuk azt, hogy az A-csoport tagjai hogyan válaszoltak az első és harmadik szakaszok megfelelő típusú kérdéseire. A 14. táblázat: Az A, B, C csoportok első szakaszbeli eredményei

(mivel a Q1.2.6 kérdés megválaszolása viszonylag magas matematikai ismeretet igényelt, ezt kihagytuk elemzésünkéből)

a diákoknak az első szakaszbeli válaszait tartalmazza. Mindenik csoport esetében átlagot számítottunk.

Érdekes megfigyelni, hogy az A-csoport eredményei, a megvizsgált algoritmusok szerinti bontásban korrelálnak azzal, hogy mennyire volt intuitív a szóbanforgó algoritmus. A bináris keresés esetében, amelynek logaritmikus az időbonyolultsága, a csoport átlagteljesítménye csak 34% volt. Ésszerű azt feltételezni, hogy a szintéziskérdés nehézségi foka a négyzetes komplexitású buborékrende­zés és a logaritmikus komplexitású bináris keresés közé esik.

Egy fontos észrevétel az, hogy bár a B-csoport tagjai tanulták mindhárom algoritmust előzőleg, nem teljesítettek következetesen jobban, mint az A-csoport tagjai. Nem volt szignifikáns különbség a két csoport között sem a lineáris keresés, sem a buborékrende­zés, sem a szintézis kérdés esetében. Sőt, a buborékrende­zéshez csatolt kérdéseken az A-csoport teljesített jobban, bár a különbség nem volt szignifikáns. Az egyetlen algoritmus, amely kapcsán a B-csoport szignifikánsan jobban teljesített az a bináris keresés, azaz a legkevesbé intuitív algoritmus. Nyilván, a C-csoport szignifikánsan jobban teljesített, mint az A is és a B is, mindhárom algoritmus esetében. A szintézis kérdés kapcsán is jobb­ak voltak, de a különbség nem volt szignifikáns.

Az, hogy a C-csoport kimagaslóan teljesített, egy kézenfekvő eredmény, hiszen a kérdések olyan témakörből voltak, amelyet a középiskolában már tanultak (algoritmusok időbonyolultsága). A másik két csoport számára viszont, ez egy ismeretlen témakör volt. Továbbá, hogy a B-csoport tagjai már ismerhették a középiskolából a megvizsgált algoritmusok stratégiáját (I. típusú ismeret), egy nyilvánvaló előnyt jelentett számukra, az A-csoport­hoz vi-

szonyítva. Igaz a kérdések II. típusú ismeretre vonatkoztak, de ezek megválaszolása az I. típusú ismeretet is feltételezte (Hogyan tudná valaki elképzelni egy olyan algoritmus legjobb és legrosszabb esetekben való viselkedését, amelynek nem érti a stratégiáját). Szembetűnő eredmény tehát, hogy mind ezek ellenére az A- és B-csoportok váll-váll mellett teljesítettek a Q1.1.1-6, Q1.3.1-2, Q1.4.1 kérdések esetében. Ez részben megválaszolja az *első* kutatási kérdésünket, azaz, hogy a megvizsgált tanulási környezetben van potenciál arra, hogy elsőéves, programozási ismeretekkel nem rendelkező diákokat bevezessen az algoritmusok világába. Az alábbiakban összefoglaltunk néhány korábbi, algoritmusvizualizáció témakörhöz kapcsolódó kutatási eredmény, amely megmagyarázhatja következtetésünket:

- a merész tudomány-művészetek társítás növelheti a diákok belső motivációját (Kátai, 2014);
- az emberi mozgás alkalmazása hatékonyabb dinamikus vizualizációkat eredményez (Juan et al., 2018);
- a csatolt kérdéssorozatok segíthették a diákokat abban, hogy a tánckoreográfiák és animációk releváns elemeire összpontosítsanak (Graesser & Person, 1994; Wu & Rau, 2018).

Ami a második kutatási kérdésünket illeti, az hogy az A-csoport alulteljesített a B-csoportéhoz képest a kevésbé intuitív algoritmus esetében, illetve hogy úgy az A mint a B-csoport szignifikánsan lemaradt a C-csoportéhoz képest, arra enged következtetni, hogy tanári segítség nélkül a környezetben rejlő potenciál csak részlegesen aknázható ki (Sweller et al., 2010). Amikor összehasonlítottuk az A-csoportnak az első és harmadik szakaszokra vonatkozó eredményeit ugyancsak erre a következtetésre jutottunk.

	Legjobb eset (hasonlítás)	Legjobb eset (csere)	Legrosszabb eset (hasonlítás)	Legrosszabb eset (csere)
Első szakasz (buborékren- dezés)	Q1.3.1 (9) 78%	– (0) –	– (45) –	Q1.3.2 (45) 40%
Harmadik szakasz (kiválasztó rendezés)	Q3.1.1 (45) 51%	Q3.1.2 (0) 95%	Q3.1.3 (45) 71%	Q3.1.4 (45) 57%

5.táblázat: Az A csoport első és második szakaszbeli eredményei

Amint a 5.táblázat: Az A csoport első és második szakaszbeli eredményei szemlélteti, az A csoport tagjainak az 57%-a ismerte fel, hogy a kiválasztó rendezési algoritmus 45 cserét végez, ha a bemenet egy csökkenő sorrendbe rendezett számsorozat (legrosszabb eset). Ez szignifikánsan jobb eredmény (t-teszt: $p=0,05$), mint amit e diákok az első szakasz hasonló típusú kérdésénél értek el (az Q1.3.2 kérdésre csak 40% válaszolt helyesen).

Egy sajátossága a kiválasztó rendezésnek az, hogy legjobb esetben is $n(n-1)/2$ összehasonlítást végez. Következésképpen, a Q3.1.1 kérdés jelentősen nehezebb volt, kevésbé volt intuitív, mint az ennek megfelelő első szakaszbeli Q1.3.1 kérdés. Ez megmagyarázhatja, hogy miért esett vissza a teljesítmény 78%-ról 51%-ra. Másfelől viszont, figyelemreméltó eredménynek tekinthető az is, hogy olyan diákoknak több mint fele, akik előzőleg nem tanultak programozást, felismerte, hogy a kiválasztó rendezés $n(n-1)/2$ összehasonlítást és 0 cserét végez legjobb esetben.

Egy további részlet, hogy a második és harmadik szakaszok keretében, a kísérleti csoport diákjai (A csoport) négy alcsoporthoz tartoztak. A Q3.1.1 adathalmaz furcsa sajátossága, hogy az egyik alcsoport teljesítménye csak 14% volt. Érdekes, hogy ha csak a többi három alcsoportot tekintjük, ezek átlaga éppen 78%, azaz ugyanannyi, mint Q1.3.1 kérdéshez kapcsolódó teljesítmény. (Mivel az első szakaszban, a buborékrendezés esetén, nem kértük a

diákoktól, hogy adják meg a cserék számát legjobb esetben, valamint az összehasonlítások számát legrosszabb esetben, ezért ezeket a szempontokat nem tudtuk bevonni az összehasonlításba).

A harmadik szakasz eredményei hangsúlyozzák a tanár irányította, második szakaszbeli, osztálytermi beszélgetés fontosságát, amely viszont az első szakasz kérdéseire épült. Mivel a harmadik szakaszban a tanulási környezet hasonló volt az első szakaszbelihez, ezért ésszerű azt feltételezni, hogy a második szakaszbeli megbeszélés jelentősen hozzájárult a tanulók harmadik szakaszbeli teljesítményéhez. Érdekes megfigyelni, hogy a résztvevők annak ellenére teljesítettek jobban a harmadik szakaszban, hogy a második szakaszbeli rávezető kérdések az első szakaszbeli algoritmusokra vonatkoztak. Az a tény, hogy a diákok képesek voltak kiterjeszteni az első szakaszbeli megértésüket egy új algoritmusra azt sugallja, hogy a második szakaszbeli megbeszélés segített a diákoknak túllátni a konkrét példákon, és általánosabb elveket is megérteni.

Mindezekkel összhangban kijelenthetjük, hogy a harmadik szakasz eredményei arra mutatnak, hogy az AlgoRhythmics környezet, amennyiben ki van egészítve egy megfelelő tanár irányította osztálytermi kérdés-felelet megbeszéléssel, jól felhasználható arra, hogy programozói előismeretekkel nem rendelkező hallgatókat rávezessünk olyan informatikai fogalmakra is, mint amilyen az algoritmusok időbonyolultsága (T4.1 tézis).

6.5. Következtetések

Ebben a fejezetben azt elemeztük, hogy miként járul hozzá az AlgoRhythmics környezetben való tanulás hatékonyságához az, ha a vizualizációkhoz jól átgondolt kérdéssorozatokot társít a tanár. E kérdések kétféle formában is megjelentek a háromszakaszos tanmenet során. Az első és harmadik szakaszok-

ban a vizualizációkhoz csatolt kérdésekben a résztvevők magukban mélyülhettek el, a második szakaszban viszont egy tanár irányította kérdés-felelet megbeszélés révén szembesültek a diákok a tanári kérdésekkel.

Az eredmények arra mutatnak, hogy az AlgoRythmics videók elég kifejezők ahhoz, hogy amennyiben megfelelő kérdéseket társítunk hozzájuk, a diákok kikövetkeztethessék belőlük a vizsgált algoritmus stratégiáját. Mivel a harmadik szakaszban, egy új algoritmuson, programozói előismeretekkel nem rendelkező diákok szignifikánsan jobban teljesítettek, mint az első szakaszban, ez arra enged következtetni, hogy egy a második szakaszbeli osztálytermi kérdés-felelet megbeszélés jelentősen hozzájárulhat a diákok megértésének elmélyítéséhez.

7. SZÁMÍTÓGÉPES GONDOLKODÁS VIZSGÁLATA KÜLÖNBÖZŐ KOROSZTÁLYÚ TANULÓKNÁL

7.1. Szakirodalmi áttekintés

Az utóbbi évtizedekben egyre nagyobb hangsúlyt kapott a számítógépes gondolkodás fejlesztésének szükségessége az oktatás minden szintjén. Értjük ez alatt az elemi, gimnáziumi, középiskolai és felsőoktatási szinteket egyaránt. Ennek érdekébe két, egymást kiegészítő irányzat kapott szárnyra: (1) kifejezetten ezt a célt szolgáló, új tantárgyak bevezetése; (2) a meglévő tantárgyakba olyan tartalmakat szőni, amelyek hozzájárulnak a diákok számítógépes gondolkodásának fejlesztéséhez. Ebben a szellemben, egyre több országban, egyre nagyobb hangsúlyt kap az, hogy minél korábban el kell kezdeni a diákok számítógépes gondolkodásának fejlesztését. Például, Manilla és munkatársai (2014) egy tanulmányt készítettek arról, hogy különböző országokban, az elemi és gimnáziumi szintű oktatásban tanuló diákok tantervei, milyen mértékben tartalmaznak olyan fogalmakat, amelyek hozzájárulnak a számítógépes gondolkodásuk fejlesztéséhez. Mindezen fokozott és céltudatos törekvések azt sugallják, hogy azok a tanmenetek, amelyek nem fókuszálnak explicite a tanulók számítógépes gondolkodásának előmozdítására, nem járulnak kellőképpen hozzá a képesség fejlődéséhez (Evans & Simkin, 1989). Ezzel összhangban számos kutatás, amelyet elsőéves egyetemi hallgatókkal végeztek, úgy következtet, hogy a diákok nem rendelkeznek azokkal a képességekkel, amire a programozás oktatásnak alapoznia kellene (Ahadi et al., 2017).

Másfelől viszont, mivel a számítógépes gondolkodás egy összetett képesség, amelynek van egy multidiszciplináris vonatkozása, egyes kutatók úgy következtetnek (Feaster, Ali, Zhai, 2014)), hogy a diákoknak, még ha nem is tanulnak kifejezetten számítógépes gondolkodás fejlesztését célzó tantárgyakat, a többi tantárgy járulékos hozadékaként fejlődhet e képességük. Ezt látszik alátámasztani a Lewandowski és munkatársai (2010) által végzett kutatássorozat, mely során a szerzők azt mérték fel, hogy programozási előismeretekkel nem rendelkező diákok, hogyan tudnak megbirkózni különböző típusú informatikai feladatokkal.

Amíg a korábbi kutatások többnyire azt elemezték, hogy egy adott korcsoporthoz tartozó diákok milyen szintű számítógépes gondolkodással rendelkeznek, jelen kutatásunkban mi azt tűztük ki célul, hogy megvizsgáljuk, észlelhető-e 3., 5., 7. és 9. osztályos tanulóknál, egyfajta fejlődés e képesség tekintetében, a jelenlegi tanmenet hozadékaként (Románia, 2018-2019).

7.1.1. Az AlgoRythmics ötletre épülő számítógépes gondolkodás teszt

Ahelyett, hogy egy általános számítógépes gondolkodási tesztet alkalmazunk, mint például amelyet Román és munkatársai (2018) dolgoztak ki, inkább úgy döntöttünk, akár csak a fentebb említett Lewandowski és munkatársai, hogy úgy mérjük le a diákok számítógépes gondolkodását, hogy egy olyan feladattal szembesítjük őket, melynek megoldása feltételezi, valamilyen szinten, e képesség birtoklását. Mivel a számítógépes gondolkodás fogalma az algoritmikus gondolkodás fogalmából nőtt ki, egy olyan tanulási környezetet hozunk létre, amely háttéréül egy számítógépes algoritmus (lineáris keresés) szolgált. A feladat megszövegezésekor odafigyeltünk arra, hogy a 3. osztályos tanulók számára is érthető és érdekes legyen. Az alapelveket, amelyet figyelembe vettünk, Comenius a következőképpen fogalmazta meg:

„A gyermekekkel megtaníthatók a felnőttkor tevékenységei, csak gyermeki módon” (Bollók, 1992).

Egészen pontosan arról volt szó, hogy a tanulóknak azonosulniuk kellett egy táncos fiú szerepével, aki a párját keresi, 7 lány között. Valójában a tanulók egy keresési problémával szembesültek, hisz a lányok egy számsorozatot képviseltek, a fiú pedig a keresett számot. A számokat úgy értelmeztük, mint amelyek tánclépéseket kódolnak. A fiúnak az a lány volt a párja, aki ugyanazt a számot viseli, azaz ugyanazon tánclépések szerint táncol, mint ő. A lányok a hátukon viselték a számokat, így a fiú, hogy megtalálja párját, arra kényszerült, hogy felkérje, sorra, egy-egy táncra a lányokat.

Ahogy a bevezetőben már utaltunk rá, főleg olyan algoritmusok tanulmányozása járul hozzá a számítógépes gondolkodás fejlesztéséhez, amelyek háttérben számítógépes modell áll (Denning, 2009). Kátai (2014) úgy következtet, hogy algoritmusvizualizációs környezetben segíthet a diákoknak a „vak” számítógép „nézőpontjával” való azonosulásban, a szelektív elrejtés elve. Ezek a szempontok vezéreltek abban a döntésünkben, hogy a lányok csak a hátukon viseljék a képviselt számokat.

Számos kutató, akik a számítógépes gondolkodás fogalmának meghatározásait elemezték, kihangsúlyozták, hogy egy komplex képességről van szó, amelynek számos összetevője van. Shute és munkatársai (2017) szerint ilyen komponensek a következők: részfeladatokra bontás, iteráció, debuggolás, absztrakció, algoritmus, általánosítás, hatékonyság stb. A párját kereső táncos fiú története mögött meghúzódó algoritmikai feladatban is fellelhetők e komponensek. Például a résztvevőknek, a keresési problémát úgy kellett felfogniuk, mint táncok sorozatát (részfeladatokra bontás). A módszer lényegében egy iteráció, amíg a keresett elem meg nem található. Az algoritmus átlátása

azt feltételezte, hogy a tanulóknak el kellett tudniuk vonatkoztatni (absztrakció) olyan elemektől, mint fiú, lányok, tánc stb. A csatolt kérdések (7.2. fejezet) kitértek bonyolultsági szempontokra is, amelyek az algoritmusok hatékonyságának fokmérői. Sőt az is megköveteltetett a tanulóktól, hogy általánosítsák válaszaikat (arra az esetre vonatkozóan, ha n lány között keres a fiú). A résztvevőknek válaszolniuk kellett ugyanarra a kérdésre, a kísérlet tanulási fázisát megelőzően is, és azt követően is. Ez felfogható úgy is, mintha lehetőségük lett volna javítani a válaszukon.

7.2. Kutatási kérdések

A jelen kutatáshoz az alábbi kutatási kérdéseket fogalmaztuk meg:

1. Észlelhető-e fejlődés abban (a jelenlegi curriculum tantárgyai hozadékaiként), ahogy 3., 5., 7. és 9. osztályos diákok (akik nem részesültek előzetes számítógépes oktatásban) viszonyulnak egy olyan feladathoz, amely egy bizonyos fokú számítógépes gondolkodást feltételez. Ha van fejlődés, akkor ez milyen mértékű?
2. Mennyire függ az esetleges korosztályonkénti növekedés a jelenlegi curriculum jellegétől: művészeti iskola vs. elméleti iskola?
3. Milyen mértékben képesek elsajátítani a különböző korosztályú tanulók egy alapvető számítógépes algoritmust?
4. Függ-e a nemektől különböző korosztályok számítógépes gondolkodásának fejlődése?

7.3. KÍSÉRLET

E kutatásra a 2018/2019-es tanév első félévében került sor a marosvásárhelyi Bolyai Farkas Elméleti Líceumban, illetve a marosvásárhelyi Művészeti Iskolában. A kísérletet ugyancsak háromszakaszosra terveztük. Az első sza-

kaszban igyekeztük felmérni azt, hogy milyen szintű lehet, az aktuális tantervük eredményeként, a résztvevők számítógépes gondolkodása, azaz milyen mértékben éreznek rá a lineáris keresési stratégiára. A második szakaszban (tanulási fázis) a tanulók megtekintették ezen algoritmus tánckoreográfia ábrázolását. A harmadik szakaszban a résztvevők olyan kérdésekkel szembesültek, amelyek feltárhatták, hogy mennyire értették meg a vizualizált algoritmust.

7.3.1. A kutatás résztvevői

A kísérlet alanyai 3., 5., 7. és 9. évfolyamos tanulók voltak, összesen 214 résztvevő (56%-a lány). Mivel Romániában elemi szinten csak elméleti és művészeti szakirány létezik, ezért esett a választás egy (e)lméleti (E-iskola) és egy (m)űvészeti (M-iskola) iskolára. Nyolc osztályt vontunk be a kísérletbe, mindenik évfolyamról, mindkét iskolából, egyet-egyet. Ezekre az osztályokra, a következőkben úgy utalunk, mint 3E, 3M, 5E, 5M, 7E, 7M, 9E és 9M osztályok. Azért kezdtük a 3. osztállyal mert Settle, Goldberg és Barr (2013) kutatók által végzett mérések arra utalnak, hogy ez az a szint, amikor a tanulók már nem olvasni tanulnak, hanem olvasással tanulnak. A további korosztályok kiválasztásánál azt feltételeztük, hogy a kétéves eltolás mérhető különbségeket eredményez, amennyiben létezik egyfajta korosztályszerinti fejlődés. E felosztás harmonizál a CSTA¹ (2017) által meghatározott szintekkel is. A 6. táblázat a résztvevők számát tartalmazza, korosztály, nemek, valamint szakirány szerinti bontásban. A résztvevők egyike sem tanult korábban programozást, így első alkalommal szembesültek egy számítógépes algoritmussal.

¹ Computer Science Teachers Association

Korosztály	Fiúk		Lányok		Összesen	
	Művészeti (M)	Elméleti (E)	Művészeti (M)	Elméleti (E)	Művészeti (M)	Elméleti (E)
3.	13	18	12	14	25	32
5.	7	16	13	13	20	29
7.	4	11	21	18	25	29
9.	9	15	17	13	26	28
Összesen	33	60	63	58	96	118

6. táblázat: A kutatásban résztvevők száma korosztályonként, nemenként, iskolánként

7.3.2. A kutatás megvalósításának eszközei

A kísérletre a tanulók által megszokott, saját oktatási környezetükben, az iskolájuk számítógépekkel felszerelt informatika laboratóriumában került sor. Kivételt képeztek a 3. osztályosok, akik számára, mivel nem szerepel a tantervükben az informatikaóra, az egyetemünk biztosította a felmérés helyszínét.

Az első és harmadik szakaszok kérdéseit a Socrative online környezetben kellett a résztvevőknek megválaszolniuk. A második szakaszban az AlgoRhythms környezet lineáris keresést ábrázoló videóját vetítettük le a diákoknak, amely flamenco tánccal illusztrálja a szóban forgó algoritmust.

7.4. Eredmények és kiértékelés

A kísérlet egy fontos sajátossága az volt, hogy nem jelöltük meg, hogy a számsorozat melyik végén kezdődjön a keresés, és a résztvevőknek sem volt lehetőségük ezt megadni. Ennek érdekes következményei lettek.

A K_1 kérdésre (Melléklet 30. ábra) helyesnek, a lineáris keresés szerinti válaszok egyikét fogadtuk el: 6 (ha balról indul a keresés) vagy 2 (ha jobbról indul a keresés). Hasonló módon, a $K_{3.1}$ kérdés (Melléklet 32. ábra:) kapcsán a 7 és 4 válaszokat tekintettük helyeseknek.

Ami a $K_{3.2.1}$ kérdést illeti (), helyes válasznak fogadtuk el, ha a megadott számsorozat 25-tel kezdődött (például: 25 1 2 3 4) vagy ha 25-re végződött (amennyiben balról indul a keresés). Mivel az sem volt kikötve, hogy a sorozat elemeinek különbözniük kell, ezért olyan helyes válaszok is születtek, amelyeknél a 25-ös szám a sorozat mindkét végén megtalálható volt. Sőt néhányan a konstans 25-ből álló számsorozatot adták válaszul. Meglepetten tapasztaltuk, hogy többen is a megadott sorozat közepén helyezték el a keresett számot (például: 1 2 25 3 4). A felmérés végén a tanulók ezt azzal indokolták, hogy mivel nem jelölhették meg, hogy hol indul a keresés, így bármelyik irányból indulna is a fiú, „legszerencsésebb” esetben van. Ezeket a válaszokat is helyesnek tekintettük.

A $K_{3.2.2}$ kérdés kapcsán a következő eseteket fogadtuk el helyesnek: a 25-ös a megadott sorozat végén, vagy elején volt, vagy nem szerepelt benne. Ez esetben is érkeztek olyan válaszok, hogy a 25-ös középen helyezkedett el. A szóban forgó diákok „legszerencsétlenebb” esetként fogták fel ezt.

A $K_{3.3.1}$ kérdés esetén az 1-et tekintettük helyes válasznak, a $K_{3.3.2}$ kérdés kapcsán pedig az x vagy $x-1$ változatokat.

A helytelen válaszok 0, a helyesek pedig 1 pontot értek. Ily módon az első szakasz feladatára 0 vagy 1 pontot kaphattak a résztvevők. A harmadik szakasz feladata kapcsán, lévén 5 alpontos, a résztvevők legtöbb 5 pontot értek el. A jó válaszokért járó pontok évfolyamonkénti átlagait (százalékban kifejezve), és a megfelelő szórás-értékeket a 7. táblázatban foglaltuk össze. Mivel a $K_{3.3.1}$ és $K_{3.3.2}$ kérdésekben az ismeretlen, az x , egy olyan matematikai fogalom volt, amelyet a 3. osztályosok még nem tanultak, ezért az elemzést elvégeztük úgy is, hogy a harmadik szakasz kapcsán csak $K_{3.1}$, $K_{3.2.1}$ és $K_{3.2.2}$ kérdésekre adott válaszokat vettük figyelembe (7. táblázat: A kutatás kérdéseire adott válaszok átlaga és zárójelben a megfelelő szórás).

	K_1	$[K_{3.1}; (K_{3.2.1}, K_{3.2.2}); (K_{3.3.1}, K_{3.3.2})]$	$[K_{3.1}; (K_{3.2.1}, K_{3.2.2})]$
3.	0,32	0,65 (0,18)	0,83 (0,22)
5.	0,43	0,80 (0,17)	0,84 (0,18)
7.	0,56	0,80 (0,24)	0,85 (0,23)
9.	0,67	0,79 (0,20)	0,89 (0,19)

7. táblázat: A kutatás kérdéseire adott válaszok átlaga és zárójelben a megfelelő szórás érték

Ahhoz, hogy megválaszoljuk a kutatási kérdéseket, azt vizsgáltuk meg, hogy van-e szignifikáns eltérés a különböző korosztályú, nemű, illetve különböző tanterv szerint tanuló diákok válaszai közt.

A K_1 kérdés kapcsán Fisher-féle egzakt tesztet alkalmaztunk, hisz az adatok bináris jellegűek voltak (helyes/helytelen válasz). A harmadik szakasz választait, t-teszt, valamint egy- és két-szemponos varianciaanalízis (ANOVA) segítségével dolgoztuk fel. Levene-próbával ellenőriztük ezek alkalmazhatóságát ($p=0,07>0,05$).

7.4.1. Korosztályonkénti elemzés

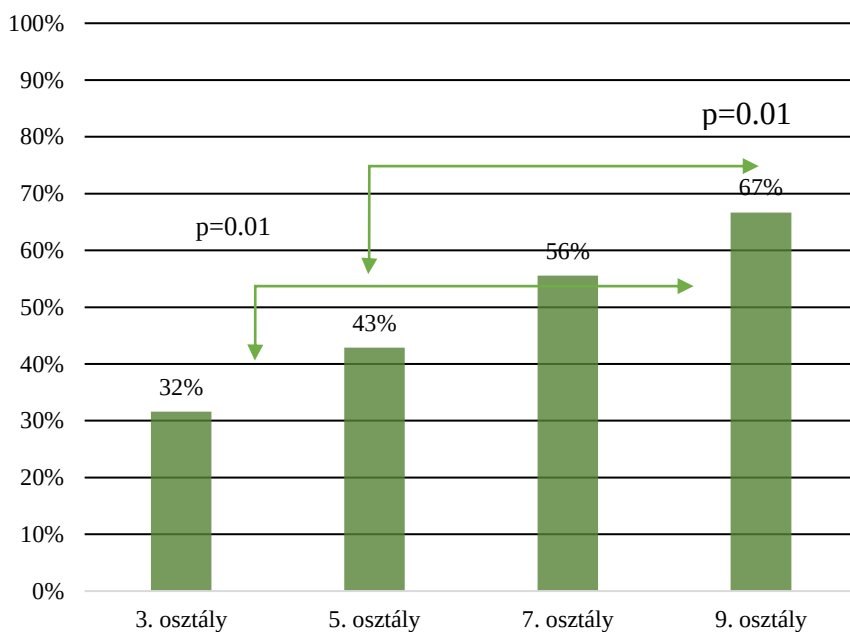
A korosztályonkénti elemzésnél a független változó a résztvevők korosztálya volt (3., 5., 7. és 9. osztály), függő változónak pedig a kérdésekre adott válaszokban tükröződő teljesítményt tekintettük. A K_1 kérdésre érkezett válaszok arra utaltak, hogy mennyire érznek rá olyan tanulók, akik programozást nem tanultak, egy számítógépes algoritmusra. A 23. ábra lineáris növekedést mutat, ami arra utal, hogy létezik egy korosztályok szerinti fejlődés a mért számítógépes gondolkodásban. A növekedés mértékének meghatározásához Fisher-féle egzakt tesztet használtunk (az adatokat 2×2 -es kontingencia-táblákba

rendeztük). Az eredmények arra mutattak, hogy nincs szignifikáns növekedés, ha a korosztályokat kétévenkénti eltolásban tekintjük (3. osztály vs. 5. osztály; 5. osztály vs. 7. osztály; 7. osztály vs. 9. osztály).

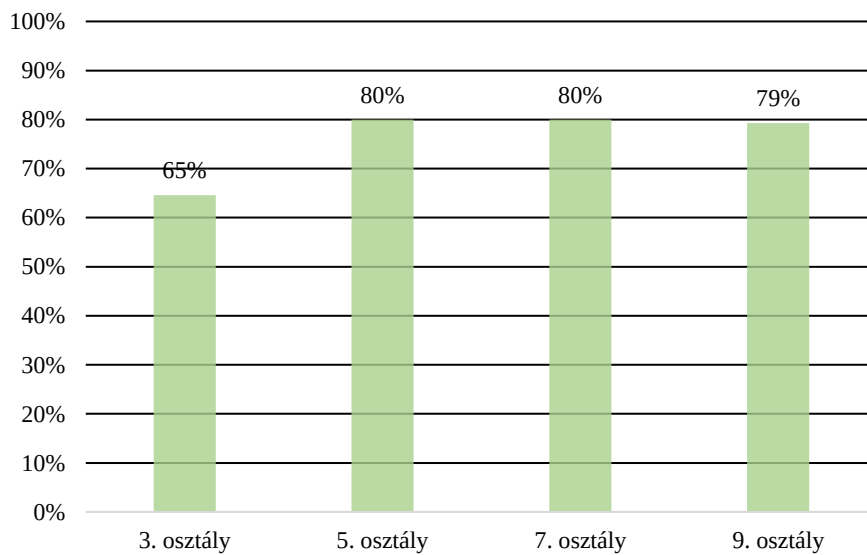
	Helyes válaszok száma	Helytelen válaszok száma
3. osztály	18	39
7. osztály	30	24
5. osztály	21	28
9. osztály	36	18

8. táblázat: Az első kérdésre adott helyes és helytelen válaszok száma osztályonként

Ezt követően négyévenkénti összehasonlítást végeztünk (3. osztály vs. 7. osztály; 5. osztály vs. 9. osztály). Ez alapján azt találtuk, hogy a 7. osztályos tanulók szignifikánsan jobban teljesítettek, mint a 3. osztályos tanulók ($p=0,01<0,05$). Hasonló eredményekhez jutottunk az 5. és 9. osztályos tanulók teljesítményét tekintve ($p=0,01<0,05$). Ezek alapján, ami az első kutatási kérdésünket illet, úgy következtethetünk, hogy a jelenlegi oktatási rendszer révén (amely nem összpontosít explicit a szóban forgó képesség fejlesztésére) a diákok számítógépes gondolkodása nem fejlődik kellő ütemben, hisz csak négyéves eltolásban mutatható ki szignifikáns növekedés (T5.1 tézis).

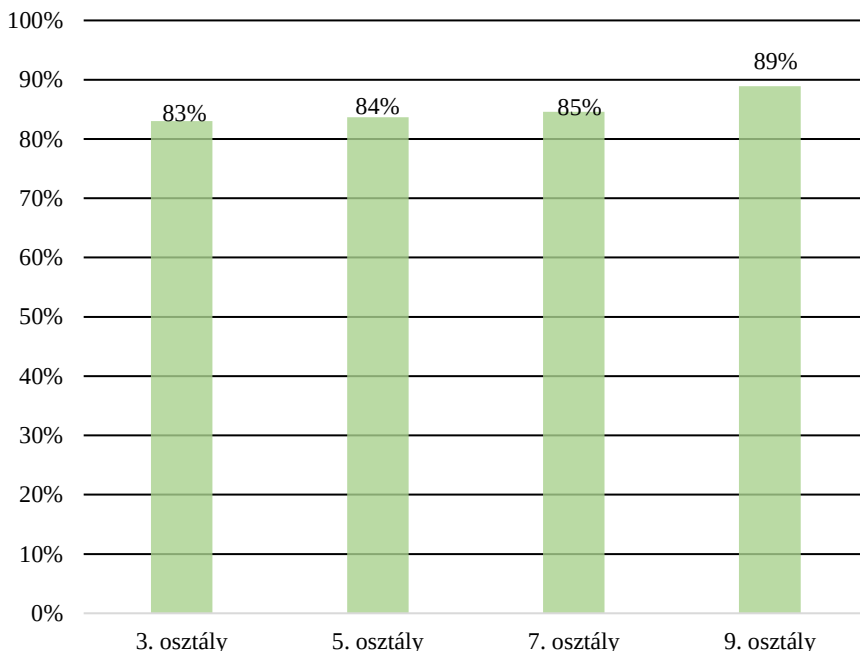


23. ábra: Az kísérlet első szakaszának eredményei korosztály szerint (mindkét iskola)



24. ábra: A kísérlet harmadik szakasz eredményei korosztály szerint (mindkét iskola)

A kísérlet harmadik szakaszában feltett kérdéssorozatra érkezett válaszokat egyszempontos varianciaanalízissel (ANOVA) elemeztük. Az eredmények azt mutatták (24. ábra), hogy szignifikáns különbség van a korosztályok közt ($F(210,3) = 7,79$, $p = 0,00$). Észrevettük azonban, hogy a különbség a 3. osztályos tanulók eredményeiből adódik (65%), hiszen az 5., 7. és 9. osztályosok váll-váll mellett teljesítettek (80%, 80%, 79%). Ezt további tesztek is alátámasztották. A $(3, -1, -1, -1)$ kontraszt értékeket alkalmazva világossá vált, hogy a 3. osztály eredményei szignifikáns különbséget mutatnak a többi osztályok válaszhoz képest ($t_{210} = 4,83$, $p = 0,00$). Tovább elemezve az adatokat nyilvánvalóvá vált, hogy ezek a korosztályú tanulók, mindkét iskolából, a $K_{3.3.1}$ és $K_{3.3.2}$ kérdésekre adott válaszaik alapján maradtak le a többi osztályokhoz képest. Ezt a lemaradást az e kérdésekben szereplő „x lány” (valamennyi lány) kifejezés okozhatta, hiszen ez a fogalom a 3. osztályosok számára még ismeretlen volt. Hogy megbizonyosodjunk erről, újra elemeztük a harmadik szakaszbeli válaszokat csak az első három kérdés alapján ($K_{3.1}$, $K_{3.2.1}$, $K_{3.2.2}$), és valóban a szignifikáns különbség eltűnt ($p = 0,47$) (25. ábra).



25. ábra: A kísérlet harmadik szakaszának eredményei, az utolsó két kérdés kivételével, korosztály szerint (mindkét iskola)

Ez megválaszolja a harmadik kutatási kérdésünket, azaz, hogy bizonyos alapvető számítógépes gondolkodásra orientált fogalmak, amennyiben megfelelő módszereket alkalmazunk, ugyanolyan hatékonysággal elsajátíthatók különböző korosztályú tanulók által (T5.2 tézis).

7.4.2. Iskolánkénti elemzés

Mivel feltételeztük, hogy a két iskola (művészeti és elméleti) tantervei közti különbségek a résztvevők válaszaiban is megmutatkozhatnak, ezért feldolgoztuk az adatokat iskolánként is. Ahhoz, hogy az elemzés helyes legyen, szem előtt kell tartanunk a következőket.

- A kísérletben résztvevő 3M osztályos tanulók a zeneszakon, az 5M, 7M, 9M osztályos tanulók a rajzszakon tanultak. A művészeti iskolában, elemi szinten (1–4. osztályok) csak zene, 5. osztálytól pedig rajz

és zene szakon tanulnak a diákok. Ebből kifolyólag az 5M osztály tagjai nem művészeti szakirány szerint végezték 1–4 osztályos tanulmányaikat.

- 9. osztályba a tanulók kísérettségi révén jutnak be. Ennek következményeként jelentős átrendeződés van az iskolák között. Például azok a tanulók, akiket vonz az informatika, 8. osztály után távoznak a művészeti iskolából, és valamelyik elméleti iskolában folytatják tanulmányaikat.

Az 9. táblázat összefoglalja, hogy a kísérletbe bevont osztályok hány évet tanultak a megfelelő iskola tanterve szerint.

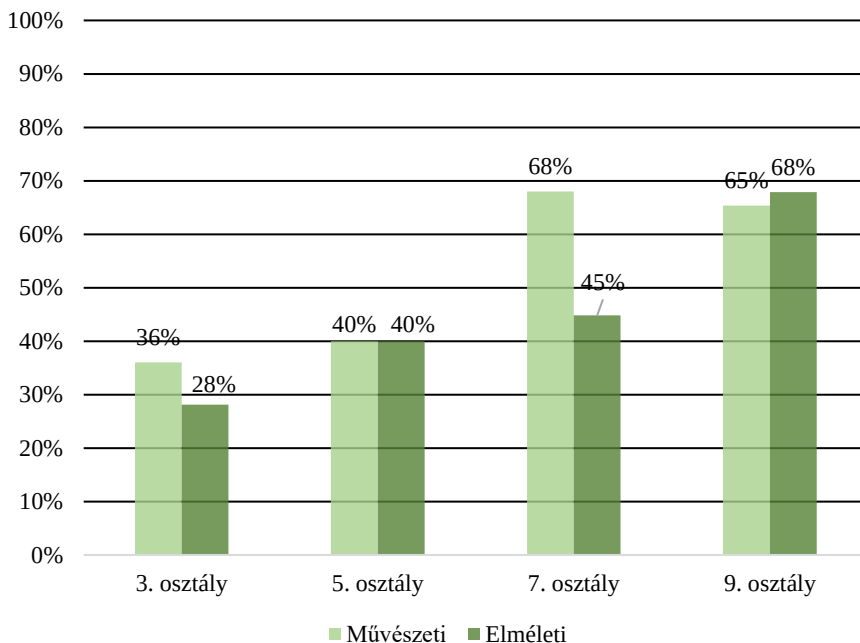
Osztály	Az M-iskolában tanult évek száma	Az E-iskolában tanult évek száma	Más iskolában tanult évek száma	Megjegyzés
3M	2 év (1–2. osztály)	–	–	Összehasonlítható a 3E-vel
3E	–	2 év (1–2. osztály)	–	Összehasonlítható a 3M-mel
5M	–	–	4 év (1–4. osztály)	Csak 5. osztálytól indul rajz szak
5E	–	4 év (1–4. osztály)	–	Többségükben az E-iskolában folytatják tanulmányaikat
7M	2 év (5–6. osztály)	–	4 év (1–4. osztály)	Összehasonlítható a 7E-vel
7E	–	6 év (1–6. osztály)	–	Összehasonlítható a 7M-mel
9M	4 év (5–8. osztály)	–	4 év (1–4. osztály)	Elmennek a reál beállítottságú diákok
9E	–	8 év (1–8.)	–	Jó képességű diákok érkeznek mindenfelől

9. táblázat: Adott iskolában tanult évek száma szakterületenként

Mindezek alapján kijelenthető, hogy az iskolák szerinti elemzéshez a legrelevánsabb összehasonlítások, a 3M vs. 3E, illetve 7M vs. 7E szembeállítások, hisz ezen osztályok tanulói legalább két évig folytonosan az adott szakirány tanterve alapján tanultak.

A 26. ábra a K₁ kérdés nyomán született eredményeket mutatja be iskolák szerinti bontásban. Amennyiben összességében tekintettük az adatokat nem találtunk szignifikáns különbséget a két iskola között (M-iskola: 53%, E-iskola: 46%; Fisher-féle egzakt teszt: $p=0,33 > 0,05$), és ez igaz volt akkor is, ha az 5. és 9. osztályos tanulók teljesítményét külön-külön vizsgáltuk. Másfelől viszont, amikor a korosztályok szerinti növekedésekre összpontosítottunk, meglepő eredményekhez jutunk.

Megfigyeltük, hogy a „3 vs. 7” eltoláshoz tartozó növekedés, a művészeti iskolában szignifikáns (Fisher-féle egzakt teszt, $p=0,04$), az elméleti iskolában viszont nem. Nyilvánvaló, hogy e növekedéshez főleg az 5.-ről 7.-re való eltolás járult hozzá (40%-ról 68%-ra). Ez az M-iskola tantervének előnyére írható, ami a számítógépes gondolkodás fejlesztését illeti. Ezzel összhangban megfigyelhető az is, hogy a művészeti iskola 3. osztályos tanulói jobban teljesítettek, mint az elméleti iskola harmadikosai (36% vs. 28%), és ugyanez volt megfigyelhető a 7M és 7E osztályok esetében (a különbségek bár nem szignifikánsak, jelentősek).



26. ábra: A kísérlet első szakaszának eredményeinek feldolgozása iskolák szerint

A második kutatási kérdésünk kapcsán a következtetésünk tehát az, hogy bár a jelenlegi oktatás mérsékelten járul hozzá a számítógépes gondolkodás fejlesztéséhez, az eredmények arra engednek következtetni, hogy a művészeti iskolákban e növekedés jelentősebb lehet, mint az általános tanterv szerint oktató elméleti iskolákban (T5.1 tézis).

7.4.3. Nemek szerinti elemzés

Végül elemeztük az adatokat a résztvevők neme szempontjából is. Ami a negyedik kutatási kérdést illeti, az eredmények arra mutatnak, hogy a fiúk és a lányok, alapvetően, egyformán jól teljesítettek (T5.2 tézis). Például, a K₁ kérdés kapcsán mindkét nem teljesítménye 49%-os volt. A harmadik szakasz kérdéssorozatán pedig szignifikánsan jobban teljesített mindkét nem, a fiúk teljesítménye 74%-ra, a lányoké pedig 77%-ra nőtt. Amikor iskolák szerinti bontásban vizsgáltuk az adatokat, hasonlóképpen nem találtunk szignifikáns

eltérést a két nem között. Érdekes viszont, hogy az első szakaszban az E-iskolában a fiúk (fiúk 52%, lányok 40%), az M-iskolában pedig a lányok (fiúk 45%, lányok 57%) teljesítettek valamivel jobban. A harmadik szakaszban a lányok eredményei valamivel nagyobb értékeket mutattak mindkét iskolában (E-iskola: fiúk 76%, lányok 78%; M-iskola: fiúk 73%, lányok 77%). Ahhoz, hogy korosztály szerinti bontásban is megvizsgáljuk a fiúk és lányok eredményeit, két-szemponos varianciaanalízist (ANOVA) alkalmaztunk. A két független változó a korosztály (3., 5., 7., 9), valamint a résztvevők neme (fiú, lány) volt, a függő változó pedig a diákok harmadik szakaszbeli teljesítménye (a Levene próba igazolta e teszt alkalmazhatóságát; $p=0,55>0,05$). Az eredmények azt mutatták, hogy nincs interakció a két független változó, azaz a korosztály és a nem között.

Mindez összhangban van egyéb, hasonló témakörben végzett kutatások következtetéseivel (Ahadi et al., 2017; Byrne & Lyons, 2001; Werth, 1986). További tanulmányok, amelyek hasonló eredményekhez jutottak, például Lau és Yuen (2009) kutatása, akik 270, 14–19 év közötti, középiskolás tanulón mértek vagy az Atmatzidou és Demetriadis (2016) tanulmánya, amely ugyancsak középiskolás fiúk és lányok számítógépes gondolkodása fejlődését vizsgálta. A mi kutatásunk kiterjeszti ezen eredményeket elemi és gimnáziumi szintre.

Másfelől azonban, ami a $K_{0.1}$ (Mennyire szereted a számítógépezést és az informatikát?) és K_2 (Mennyire értetted meg a videón látott fiú alkalmazta keresési stratégiát?) kérdéseket illeti, a fiúk pontszámai szignifikánsan nagyobbak voltak, mint a lányokéi. Az összehasonlításhoz kétmintás t-próbát használtunk, ahol független változó a tanulók neme, a függő változó pedig a megfelelő kérdéshez tartozó pontszám volt (zárójelben az átlagokat és a megfelelő szórás értékeket adtuk meg):

- $K_{0.1}$ kérdés: fiúk (8,77, 1,92); lányok (7,73, 2,14); $t(212)=3,68$,
 $p=0,0002<0,05$
- K_2 kérdés: fiúk (8,24, 2,22); lányok (7,40, 2,51); $t(212)=2,52$,
 $p=0,01<0,05$

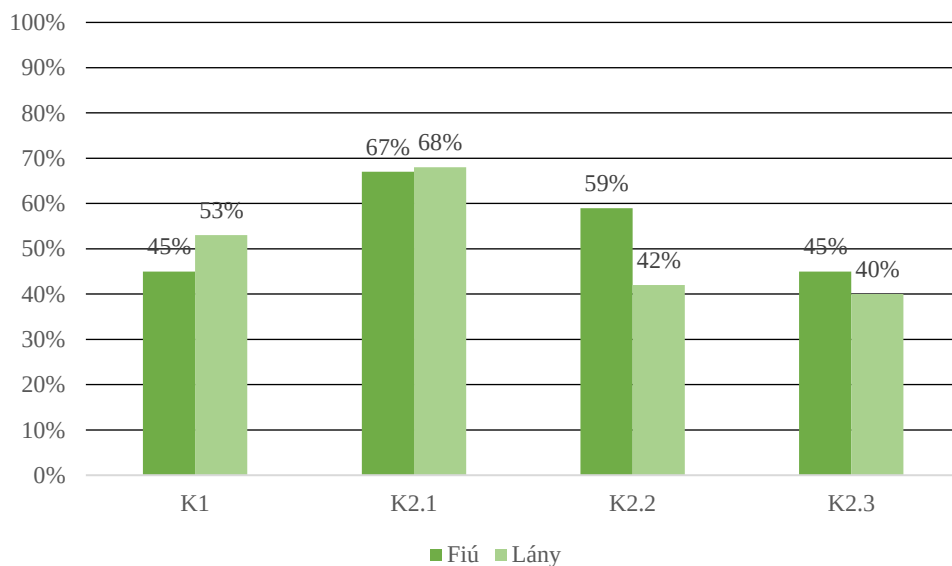
Ezek az eredmények összhangban vannak azokkal a korábbi kutatásokkal, amelyek a lányok önbizalom hiányát igazolják ebben a témakörben, képességeik és sikereik ellenére (Ahadi et al., 2017; Stout & Tamer, 2016).

A $K_{0.2}$ kérdés (Mennyire szereted a zenét és a táncot?) kapcsán a lányok válaszaik mutattak magasabb értékeket.

7.5. Következtetések

Ebben a kutatásban azt vizsgáltuk, hogy 3., 5., 7. és 9. osztályos tanulóknak milyen ütemben fejlődik a számítógépes gondolkodása. A kísérlet alanyai két különböző iskolából származtak, egy elméletiből és egy művészetiből. A résztvevők egyike sem tanult korábban olyan tartalmakat, amelyek kifejezetten a számítógépes gondolkodásuk fejlesztését célozta volna. Az eredmények azt mutatták, hogy a jelenlegi tanterv tantárgyainak járulékos hozadékeként létezik egyfajta növekedés, de ez viszonylag lassú. Kétévenkénti eltolásban nem mutatható ki, viszont négyévenkéntiben igen. Egy érdekes észrevételünk, hogy úgy tűnik, hogy a művészet iskola tanterve inkább hozzájárul a diákok számítógépes gondolkodásának fejlődéséhez, mint az elméleti iskoláé.

Egy további következtetése a jelen fejezetnek az, hogy megfelelő módszerekkel és eszközökkel a különböző korosztályú diákok egyformán jól megtaníthatók olyan alapvető, számítógépes gondolkodáshoz kapcsolódó fogalmakra, mint amilyen a lineáris keresési algoritmus. Az eredmények azt is igazolták, hogy e tekintetben nincs különbség fiúk és lányok között (27. ábra).



27. ábra: A kutatási kérdésekre adott válaszok feldolgozása nemenként

8. LIMITÁCIÓK

Mindenik kutatás limitációjának tekinthető, hogy csak egy-két algoritmuson történtek a mérések. Megjegyzendő viszont, hogy már az AlgoRhythmicss kollekció is lehetővé teszi ezek kiterjesztését akár 10 algoritmusra is, hiszen ennyi vizualizáció már rendelkezésre áll. Egy másik szempont, hogy a jelen kutatásokban eltekintettünk a motivációs szempontoktól, viszont elképzelhető, hogy a motivációs elemek is befolyásolták az eredményeket.

Ami az első két kutatást illeti az is korlátként fogható fel, hogy az AlgoRhythmicss videók nagyon sajátos realisztikus ábrázolások, ezért az eredmények általánosítása, bármely realisztikus vizualizációra, nem magától értődő. Ide kapcsolódó limitáció az is, hogy a videók is tartalmaztak néhány absztrakt elemet, hisz a számsorozatok tároló tömböket gyakran téglalapok képviselték.

Egy további szempont, hogy a számítógépes gondolkodás egy összetett képesség, amelynek számos komponense van, amelyek közül mi általában csak kettőre összpontosítottunk, az algoritmus és algoritmus-hatékonyság fogalmakra.

9. ÖSSZEGZÉS

Ebben a dolgozatban különböző korosztályú tanulók számítógépes gondolkodása fejlesztésének lehetőségeire összpontosítottunk interaktív, többérvű-szerves, elektronikus oktatási környezetekben. Egészen pontosan azt vizsgáltuk, hogy miként járulhatnak ehhez hozzá az AlgoRhythmic környezetbe ágyazott algoritmusvizualizációk.

9.1. Új fejlesztések

Az AlgoRhythmic fejlesztés 2007-ben indult, viszont 2016-ban új lendületet kapott. Ekkortól vagyok jómagam is része a kutatócsoportnak, és a dolgozatban bemutatott kutatások is azóta zajlottak. Az első tánckoreográfiákat 2011-ben tette közzé a kutatócsoport. Ezek hat számítógépes algoritmust illusztráltak: kiválasztó, beszűrő, buborék, Shell, összefésülő és gyors rendezés. A nemzetközi visszhang nagyon kedvező volt, és rengeteg kérés érkezett a kupacrendezés vizualizációjának elkészítésére, illetve arra, hogy terjesszük ki a módszert más algoritmuscsoportokra is. Az én feladatom volt, hogy főszerepet vállaljak ebben. Sikerült számos új elemmel gazdagítani az azóta született négy új algoritmus ábrázolást (kupacrendezés, lineáris és bináris keresés, visszalépéses keresés a négykirálynő-problémára). Például, az előző videókban mindig egy-egy táncos képviselte a számsorozat elemeit, a kupac rendezés illusztrációban viszont táncos párok töltik be a szerepet. Egy további újdonság ebben a videóban, hogy a számsorozat egydimenziós ábrázolása mellett megjelenik a kétdimenziós ábrázolás is, hiszen a kupac úgy mond kinyílik bináris fává. Új tánc stílusokat vontunk be, ugyanis a lineáris és bináris keresések flamenco tánccal, a visszalépéses keresési algoritmus pedig balett koreográfiával van illusztrálva. Az előző kutatások arra mutattak, hogy növel-

heti a vizualizációk hatékonyságát, ha a számsorozatok „rejtetten vannak ábrázolva”. Ennek következményeként a lineáris és bináris keresések esetében a számsorozatot képviselő lányok csak a hátukon viselik a számokat.

A kutatócsoport már a kezdetektől tudatában volt, hogy a tánckoreográfia illusztrációk nem helyettesítik az algoritmusok absztrakt animációkkal való ábrázolását. Az is világos volt, hogy ahhoz, hogy a vizualizációkban rejlő potenciál kellőképpen kiaknázható legyen, szükséges egy megfelelő tanítási-tanulási környezetet építeni köréjük. Már 2012-ben készült egy webes alkalmazás e célból, amely a videók mellé számítógépes animációkat társított, és lehetővé tette az ezekkel való interaktív tanulást. Ahogy készültek el az újabb algoritmus illusztrációk, úgy vált nyilvánvalóvá, hogy az AlgoRythmics webes környezet is megújításra szorul. Néhány fejlesztés, amelyben kulcsszerepet vállaltam a következők: legyen beállítható, hogy milyen interaktivitási szinttel szeretné a környezetet a felhasználó használni; nyújtson segítséget a környezet az algoritmus megértése és a számítógépes kód elkészítése közt tárgyongó szakadék áthidalásához; tegye lehetővé a környezet a felhasználók teljesítményének mérését.

9.2. Új kutatások

A 2016 előtt megvalósított kutatások többnyire az AlgoRythmics környezet többérzékszerves jellegét vizsgálták. A kapcsolódó publikációk címe is ezt tükrözik: „On the role of senses in education” (2008), „Technologically and artistically enhanced multi-sensory computer-programming education” (2010), „Multi-sensory method for teaching-learning recursion” (2011), „Multi-Sensory Informatics Education” (2014) stb. Azt követően, hogy bekapcsolódtam a projektbe, új szempontok kerültek előtérbe.

Új nézőpontból tekintve a kétfajta AlgoRhythmics ábrázolásra, világossá vált, hogy a tánckoreográfiák a realisztikus, a számítógépes animációk pedig a sematikus dinamikus ábrázolások kategóriájához köthetők. Az egyik kutatási vonal, amelyet követtem tehát az volt, hogy e szempontból tanulmányozzam az AlgoRhythmics vizualizációkat. A szakirodalmi áttekintés után világossá vált, hogy a tánckoreográfiák egyik erőssége az lehet, hogy emberi mozgással ábrázolják az algoritmus lépéseit. A leggyakoribb ok, amiért a korábbi kutatások többnyire a sematikus ábrázolások javára következtetnek, hogy ezek kevesebb figyelemelterelő elemet tartalmaznak. Az emberi mozgás központi szerepéből kiindulva, kétféle figyelemelterelő elemet különböztettem meg az AlgoRhythmics videókban: reprezentációst, amikor a művészi jegyek árnyékot vetnek az algoritmus kulcsműveleteinek ábrázolására, és dekorációs, amelyeknek csak díszítő és motiváló szerepük van. Mivel az AlgoRhythmics környezetben mindkét típusú ábrázolás elérhető (sematikus és realisztikus), ezért felmerült az a kérdés is, hogy ezek különböző módokon való kombinálása hatékonyabb tanulást eredményez-e. A két kutatás révén (3. és 4. fejezeteket), amelyeket ebben a témakörben végeztem, az alábbi téziseket sikerült bizonyítani:

[T1.1] Azok a realisztikus algoritmusvizualizációk (mint például az AlgoRhythmics tánckoreográfiák), amelyek emberi mozgás révén illusztrálják az algoritmikai tartalmat, ugyanolyan hatékonyak, sőt hatékonyabbak is tudnak lenni, mint a megfelelő sematikus ábrázolások (absztrakt számítógépes animációk).

[T1.2] Könnyebb különbséget tenni a megtekintett algoritmusok stratégiái között, amennyiben ezeket sajátos realisztikus emberi mozgást tartalmazó vizualizációk ábrázolták, mintha sematikus ábrázolások lettek volna alkalmazva.

[T1.3] Az olyan realiztikus algoritmusvizualizációk esetében, amelyek emberi mozgást alkalmaznak, főleg azok a figyelemelterelő elemek kerülendők, amelyek árnyékot vetnek azokra a mozgásokra, amelyek az algoritmus kulcsműveleteit illusztrálják (kevésbé jelentenek gondot az esetleges dekorációs figyelemelterelők).

[T2.1] Amennyiben ismételt megtekintést alkalmazunk, az emberi mozgást effektus hasznosító realiztikus algoritmusvizualizációk (mint például az AlgoRythmics videók) hatékonyabbak, mint a megfelelő sematikusak ábrázolásai (például absztrakt számítógépes animációk).

[T2.2] Amennyiben sematikus és realiztikus algoritmusvizualizációt is alkalmazunk, előnyösebb lehet, ha előbb a sematikus, majd a realiztikus kerül megtekintésre. Mivel a realiztikus vizualizációk könnyebben felidézhetők, mint a sematikusak, ezért ismételt megtekintés, illetve több ábrázolás kombinálása esetén előnyös, ha a tanmenet realiztikus ábrázolásra végződik.

Mivel a doktori témám egyik főpontja az interaktivitás szerepének a vizsgálata volt, ezért egy következő kutatásban azt vizsgáltam, hogy nulla (a vizualizáció zavartalan megfigyelése), fél (előre nem ismert időpillanatokban megáll a vizualizáció, és a tanuló meg kell mondja a következő lépést), vagy teljes interaktivitás (a tanulónak le kell vezényelnie a teljes algoritmust) vezet a tanulmányozott algoritmus stratégiájának jobb megértéséhez. A szakirodalommal összhangban az igazolódott be (5. fejezetben bemutatott kutatást), hogy:

[T3.1] A előzetes programozási ismeretekkel nem rendelkező (kezdők – 0 év programozás) és kevés programozási tapasztalattal rendelkező (középhaladók – 1 vagy 2 év programozás) diákok javára válhat az algoritmus-vizualizáció interaktív bemutatása. A haladók (3 vagy 4 év programozás) otthonosan mozognak bármely interaktivitási szinttel rendelkező tanfolyam esetén.

[T3.2] Nincs szignifikáns különbség a különböző interaktivitási szintekkel tanuló fiúk és lányok eredményei között

[T3.3] Ha az interaktivitásból adódóan nagyon feldarabolódik az algoritmusvizualizációs folyamat, akkor ez megnehezítheti az algoritmus stratégiájára való globális rálátást.

A saját oktatói tapasztalatomból, és más kollégák visszajelzéseiből is az derült ki, hogy amennyiben a diákok magukban használják az AlgoRhythmics környezetet, csak részben tudják javukra fordítani az ebben rejlő potenciált. Például a számítógépes gondolkodás fogalmát körüljáró definíciók hangsúlyozzák, hogy a számítógépes algoritmusok megértésén túl, ez a képesség magába foglalja azt is, hogy a tanulók képesek ezek egymáshoz viszonyított hatékonyságát is érzékelni. Ezért vizsgálódásom arra is kitért, hogy amennyiben kifejező algoritmusvizualizációk mellé jól átgondolt tanári kérdéssorokat társítunk, képesek-e a tanulók akár algoritmus bonyolultsági szempontokat is kihámozni az ábrázolásokból. A 6. fejezetben bemutatott kutatás nyomán a következő következtetésekre jutottam:

[T4.1] A tanár által irányított kérdés-felelet megbeszélésnek döntő szerepe lehet az algoritmusvizualizációs környezetekben rejlő potenciál teljesebb kiaknázásában.

A előbbi mérések mind reálszakos, elsőéves egyetemi hallgatókon történtek. Az egyik legnagyobb kihívást az jelentette számomra, hogy kiterjesszem a vizsgálatot elemi, gimnazista és középiskolás tanulókra is. E kutatás érdekes következtetésekhez vezetett, ami a jelenlegi romániai oktatási rendszer alkalmazta tantervek hozzájárulását illeti, a tanulók számítógépes gondolkodásához. Öröndetes, hogy időközben bevezették, hogy már 5. osztálytól tanulhassanak a diákok algoritmikát.

[T5.1] Ha egy oktatási rendszer nem fókuszál explicite a számítógépes gondolkodás fejlesztésére, akkor a többi tantárgyak járulékos hozadékaként csak moderáltan fejlődik e képesség.

[T5.2] Megfelelő módszerekkel és eszközökkel már elemi szinten elsajátíthatók a számítógépes gondolkodás bizonyos elemei (ugyanazon hatékonysággal, mint felsőbb osztályokban), mindkét nem esetében.

10. SUMMARY

In this dissertation we focused on the development of computational thinking in case of students from different age groups. The main concepts that we used were: interactivity, multisensorial and e-learning environment. More precisely, we carried out research based on the effectiveness of the AlgoRhythmics visualizations.

10.1. New results:

The AlgoRhythmics Universe is under construction since 2007. I started to work in the AlgoRhythmics research group since 2016 when our main goal was to improve this learning tool with new ideas and to carry out research based on the new implemented concepts. Dance choreographies has been the center of the AlgoRhythmics universe which has been published in 2011. These illustrate the algorithm visualizations of six sorting strategies (selection-, insertion-, bubble-, shell-, merge- and quick sort). These visualizations have become very popular, the audience even asked to extend these to other algorithms as well (for example heap sort). My task was to introduce new types of visualizations from many points of view. During the last few years we extended our project with new dimensions. In case of the first six dance choreography videos each element from the sequence was represented by a single dancer. As a new concept, in case of the heap sort algorithm we decided to represent each item with a dancing couple. Furthermore, we introduced the two-dimensional representation. While previously the number sequence was designed with the help of a one-dimensional array, this time we used the binary tree representation. Last but not least in order to improve the

artistic nature of the project we involved new dance styles. In case of the linear and binary search we used flamenco dance, while in case of backtracking algorithms ballet.

We have also investigated the effectiveness of visualization taking in consideration the display mode. Our results suggested that not seeing the value of the specific element from the array can increase the positive returns of this strategy. Therefore, we designed the linear and binary search algorithms so that all of the girl dancers wore their numbers on their back. Therefore, the value that the boy was looking for was hidden.

Our research group was conscious about the fact that these dance choreography videos cannot substitute commonly used abstract animations. Furthermore, we also knew that an online learning tool should be developed where all of these learning strategies can be included. The first application has been published in 2012, which included dance choreography videos and animations. After a while, this learning tool has improved. Now it contains 10 algorithms, sorting and searching strategies as well which are built around five different learning steps: video, animation, in control, create code and alive code. The level of the interactivity can be set by a teacher and students can choose from various algorithm representations as well. In addition to this, the online learning environment also gives the possibility to track students' responses from which we could easily find out the performance scores of a course.

Having this online learning tool, most of the research was conducted with the help of the AlgoRythmics learning environment. We investigated the effectiveness of multi-sensorial education. One of my main tasks was to examine the benefits of the dance choreography videos (realistic visualization) and abstract animations (schematic representation). Previous research in this field

confirmed that dance choreography videos can be very interesting and artistically enhanced because of human movement effect and abstract animations with their realistic characteristic play an important role when it comes to education and visualization. Since the AlgoRhythmics learning tool contains both of these representations, it provided an excellent opportunity to investigate the effectiveness of these two representations. The research based on these concepts was carried out based on the following thesis:

[T1.1] Students assigned to the “realistic condition” will not perform less well than their counterparts from the “schematic condition group” (for example, because of the HME);

[T1.2] For the members of the “realistic condition group” it will be easier to distinguish between the two algorithms (since as realistic visualizations, they can more easily be remembered);

[T1.3] It is not so much the decorative elements, but rather the representational distractors that can obstruct the comprehension of the relevant movements of the realistic algorithm visualizations.

[T2.1] In case of repeatedly viewing the same visualization, HME enriched visualizations (dance videos) are more effective than schematic representation (abstract animations).

[T2.2] We assumed that watching the schematic visualization first would help interpreting the realistic one (cf. constraining function of MERs, Ainsworth, 2006) and, consequently, would contribute to better learning outcomes than the realistic+schematic condition.

Besides these, interactivity and user interaction play an important role in education. Furthermore, we investigated the effectiveness of the visualization taking in consideration the level of interactivity, more precisely, the extent the

user should be involved in the learning process. Based on these, we defined three interactivity levels: no-interactivity (where users are only independent observers), half-interactivity (where users are involved in the learning process at some predefined key moments, and the need to answer questions), full-interactivity (where users need to control the animation, to “orchestrate” the algorithm).

[T3.1] Interactive visualizations can benefit beginner (no prior programming knowledge) and intermediate (1-2 years of programming) students’ results. Advanced (3-4 years of programming) students can understand the main concepts of the algorithm, regardless of the type of representation.

[T3.2] There is no significant difference between boys and girls.

[T3.3] There is a similarity between the learning style and question style. Full interactivity (C group) is a disadvantage in case of complexity questions.

Going forward, we also found out that the presence of a teacher can be very useful when we want to increase the learning outcome. If we add useful instructions to these representations we could take advantage of the potential of the learning environment and we can assure a better understanding especially in case of algorithm complexity.

[T4.1] Blended learning has an important role in order to promote the development of algorithm visualization environments.

In case of our research we examined mostly first year undergraduate students. The most challenging part of my tasks was that I needed to extend these to be available for students from elementary and high school as well.

Our research has led to interesting conclusions regarding the contribution of curriculum applied by the current Romanian education system and we are

grateful because of the fact that students can improve their algorithmic thinking from fifth grade.

[T5.1] The rate and pace of CT growth depend on the nature of the current curriculum (Arts vs. Theoretical schools).

[T5.2] We propose a new testing tool (built around the linear search dance choreography that was recently added to the AlgoRhythmics collection) for providing some measure of the CT skills of young school students. We have examined the pace at which some CT skills of young students (girls vs. boys from Art vs. Theoretical schools) latently develop as they advance with their current CS free curriculum. We have investigated the extent to which students of different ages can assimilate a basic computer algorithm (linear search).

11. KÖVETKEZTETÉSEK

Végezetül elmondhatom, hogy mély megelégedéssel tölt el, hogy az Al-goRhythmics történetének részese lehettem, és számottevően hozzá tudtam járulni ezen egyedi oktatási környezet fejlesztéséhez. Az e dolgozatba tömörített kutatások jelentősen hozzájárultak a szakmai fejlődésemhez, és bízom benne, hogy kutatási eredményeim hasznos irányelvül szolgálnak majd mindazon fejlesztők számára, akik interaktív online környezeteket terveznek a számítógépes gondolkodás fejlesztése céljával, illetve azon oktatók részére is, akik ilyen környezeteket használnak e cél eléréséhez. Azt is remélem, hogy következtetéseim kiindulópontként szolgálhatnak további kutatásokhoz.

12. KÖSZÖNETNYILVÁNÍTÁS

Mindenekelőtt megszeretném köszönni témavezetőmnek, Dr. Fazekas Gábornak, meleg és mindig biztató szavait, a bizalmát bennem és türelmét hozzám.

A debreceni kollégák, közül köszönöm Dr. Csernoch Máriának a kutatás és a dolgozat megírásához nyújtott fáradhatatlan segítségét.

Nagyon hálás vagyok az AlgoRhythmic kutatócsoport vezetőjének, Dr. Kátai Zoltánnak, aki a négy év alatt szakmailag irányított, a kutatáshoz szükséges feltételeket biztosította és kritikai megjegyzéseivel folyamatosan és kitartóan segítette az előrehaladásomat. Segítsége nélkül ez a disszertáció nem készülhetett volna el.

Nagyra értékelem a kutatócsoport korábbi tagjainak munkásságát (Tóth László, Adorjáni Alpár Károly), hisz eredményeikre építve végeztem kutatómunkámat. Fontosnak tartom megemlíteni a jelenlegi tagok közül Osztían Pálma Rozáliát, Nagy Eszter Jáhelt, Cozma Cristiánt, akik, miután a környezet újra lett tervezve, oroszánrészt vállaltak és vállalnak ennek implementálásában.

Köszönöm a Sapientia Egyetem hallgatóinak, a Bolyai, Művészeti líceum diákjainak és tanárainak, akik részt vettek a kísérletekben és beleegyezésüket adták az adatok feldolgozásához.

Végül, de nem utolsó sorban köszönöm szeretett családomnak, férjemnek, gyerekeimnek kitartó támogatásukat, türelmüket és szeretetüket. Szívből köszönöm a drága lányomnak segítségét és mindig biztató szavait.

13. MELLÉKLETEK

13.1. A kutatások menete

13.1.1. Első kutatás menete

Az előtesztre az egyetem nagy amfiteátrumában került sor. A hallgatók, a Socrative alkalmazást használva, mobilkészülékeik segítségével válaszoltak a kérdésekre. Az előzetes programozói ismeretekről szóló kérdések magukba foglalták, hogy hány évet tanult az illető résztvevő programozást a középiskolai tanulmányai alatt, illetve, hogy tanulta-e a buborék és kiválasztó rendezéseket. A számítógépes gondolkodás teszt négy-négy kérdést tartalmazott a két kiválasztott e-hód feladatokra vonatkozóan.

A tanulási szakasz első felében a résztvevők megtekintették a buborékrendezési algoritmusnak vagy az elanimált (S-csoport) vagy az eltáncolt változatát (R-csoport). Ezt követte a kiválasztó rendezés vizualizációja, az S-csoport részére animációval, az R-csoport számára pedig tánckoreográfiával. A csoportok egymást követően tekinthették meg a két vizualizációt, egy-egy amfiteátrumban, a tanár laptopjáról kivetítve. Mindenik vizualizáció 10 elemű véletlen-számsorozaton illusztrálta az algoritmust. Odafigyeltünk, hogy az animációknak olyan sebességet állítsunk be, hogy ugyanazon algoritmus kétfajta vizualizációjának ugyanannyi legyen a hossza.

Az utóteszt kérdéseinek megválaszolásához újra a Socrative alkalmazást használták a résztvevők. Mindkét rendezési algoritmus kapcsán 13 kérdést kellett megválaszolniuk (a kérdések szándékosan más hosszúságú számsorozatokra vonatkoztak, mint amilyenek a vizualizációk lettek bemutatva).

- 1–4 kérdések: Adott a következő számsorozat: 12 7 15 5 6 4 8. A számsorozat elemeit az a tömbben tároljuk és 0-tól sorszámozzuk. A

Buborékredezés algoritmus alapján, melyik az első menet első/második/harmadik/negyedik művelete: Hasonlít($a[\dots]$, $a[\dots]$) /

Csere($a[\dots]$, $a[\dots]$)? Írd be a helyes választ szóköz nélkül, a pontokat helyettesítsd a megfelelő számokkal.

- 5–8 kérdések: Adott a következő számsorozat: 12 7 15 5 6 4 8. A számsorozat elemeit az a tömbben tároljuk és 0-tól sorszámozzuk. A Kiválasztó rendezés algoritmus alapján, melyik az első menet első/második/harmadik/negyedik művelete: Hasonlít($a[\dots]$, $a[\dots]$) / Csere($a[\dots]$, $a[\dots]$)? Írd be a helyes választ szóköz nélkül, a pontokat helyettesítsd a megfelelő számokkal.
- kérdés: Adott a következő számsorozat: 12 7 15 5 6 4 8. A számsorozat elemeit az a tömbben tároljuk és 0-tól sorszámozzuk. A Buborékredezés algoritmus alapján, melyik a második menet első művelete: Hasonlít($a[\dots]$, $a[\dots]$) / Csere($a[\dots]$, $a[\dots]$)? Írd be a helyes választ szóköz nélkül, a pontokat helyettesítsd a megfelelő számokkal.
- kérdés: Adott a következő számsorozat: 12 7 15 5 6 4 8. A számsorozat elemeit az a tömbben tároljuk és 0-tól sorszámozzuk. A kiválasztó rendezés algoritmus alapján, melyik a második menet első művelete: Hasonlít($a[\dots]$, $a[\dots]$) / Csere($a[\dots]$, $a[\dots]$)? Írd be a helyes választ szóköz nélkül, a pontokat helyettesítsd a megfelelő számokkal.
- 11–14 kérdések: Egy 5 elemű, szigorúan növekvő/csökkenő számsorozat esetén (best case, worst case), hány összehasonlítás/csere történik, a buborékredezés algoritmus alapján?
- 15–18 kérdések: Egy 5 elemű, szigorúan növekvő/csökkenő számsorozat esetén (best case, worst case), hány összehasonlítás/csere történik, a kiválasztó rendezés algoritmus alapján?

- 19–22 kérdések: Egy n elemű, szigorúan növekvő/csökkenő számsorozat esetén (best case, worst case), hány összehasonlítás/csere történik, a buborékrende­zés algoritmus alapján?
- 23–26 kérdések: Egy n elemű, szigorúan növekvő/csökkenő számsorozat esetén (best case, worst case), hány összehasonlítás/csere történik, a kiválasztó rendezés algoritmus alapján?

Kérdőív részei	Tartalmi egységek	Kérdések száma
	Előteszt	
blokk	Adatvédelmi irányelvek	1
blokk	Demográfiai adatok	3
blokk	Előzetes programozási ismeret	5
blokk	Számítógépes gondolkodás teszt	8
	Utóteszt	
blokk	Demográfiai adatok	2
blokk	Buborékrende­zés (1. menet)	4
blokk	Kiválasztó rendezés (1. menet)	4
blokk	Buborékrende­zés (2. menet)	1
blokk	Kiválasztó rendezés (2. menet)	1
blokk	Buborékrende­zés (bonyolultsági kérdések – ismert az elemek száma)	4
blokk	Kiválasztó rendezés (bonyolultsági kérdések – ismert az elemek száma)	4
blokk	Buborékrende­zés (bonyolultsági kérdések és általánosítás – ismeretlen az elemek száma)	4
blokk	Kiválasztó rendezés (bonyolultsági kérdések és általánosítás – ismeretlen az elemek száma)	4

10. táblázat: Kérdőív szerkezete (Sematikus versus emberi mozgást alkalmazó realisztikus algoritmus-vizualizáció)

13.1.2. Második kutatás menete

Az előtesztre egy amfiteátrumban került sor, az összes résztvevő számára egyszerre. A két algoritmus pszeudokódját papíron kapták meg a diákok. A kérdések megválaszolása viszont online történt, a Socrative alkalmazás segítségével, a résztvevők mobiltelefonjáról. Mindkét algoritmusra vonatkozóan a kérdések az alábbiak voltak (a hallgatók 0–20 közötti pontszámot érhetnek el):

- 1–2 kérdések: Tekintsük az 5 3 9 1 7 ötelemű számsorozatot. Hány összehasonlítás/cseré történik a megadott buborék/kiválasztó rendezési algoritmus során?
- 3–6 kérdések: Ha az x tömb egy ötelemű szigorúan növekvő/csökkenő számsorozatot tárol, akkor mennyi az összehasonlítások/cserék száma a megadott buborék/kiválasztó rendezési algoritmus során?
- 7–10 kérdések: Ha az x tömb egy n -elemű szigorúan növekvő/csökkenő számsorozatot tárol, akkor mennyi az összehasonlítások/cserék száma a megadott buborék/kiválasztó rendezési algoritmus során?

A tanulási szakaszhoz a csoportokat külön termekbe irányítottuk, ahol minde-nik a megfelelő tanmenet szerint tekinthette meg a Shell-rendezési algoritmust (28. ábra: A Shell-rendezés realisztikus ábrázolás, 29. ábra: A Shell-rendezés sematikus ábrázolás). A levetítésekre a tanár laptopjáról került sor. Mivel egyik diák eltévesztette a termet, ezért négy 21 tagú csoport helyett, a csoportok 20, 21, 21 és 22 létszámúak lettek. A csoportokhoz az alábbi tanulási feltételeket társítottuk:

- S-S csoport: 2-szer tekintette meg a sematikus vizualizációt, egymást követően.

- R-R csoport: 2-szer tekintette meg a realisztikus vizualizációt, egymást követően.
- S-R csoport: először a sematikus, majd a realisztikus vizualizációt tekintette meg, egymást követően.
- R-S csoport: először a realisztikus, majd a sematikus vizualizációt tekintette meg, egymást követően.



28. ábra: A Shell-rendezés realisztikus ábrázolása



29. ábra: A Shell-rendezés sematikus ábrázolása

Az utóteszt közvetlen a tanulási szakasz után következett. A kérdések feltevéséhez és megválaszolásához újra a Socrative online környezetet használtuk. A kérdéssor megbízhatósága elfogadhatónak bizonyult (Cronbach $\alpha = 0,58$). Mivel a tanulási fázisban a tánccoreográfia 10 elemű számsorozaton, az animáció pedig 12 elemű számsorozaton illusztrálta az algoritmust, ezért az utóteszt kérdések egy 7 elemű számsorozatra vonatkoztak, annak érdekében, hogy egyik csoportot se részesítsük előnyben. A számsorozatot, amelyre az alábbi kérdések vonatkoztak az x tömb tárolta ($x[1 \dots 7] = \{1, 19, 7, 8, 12, 11, 9\}$), a növekmény („gap”) értékek 3 és 1 volt, és 0, illetve 14 közötti pontszámot lehetett elérni (11. táblázat).

- 1–6 kérdések: Add meg az első három műveletet (hasonlít/csere ($x[?], x[?]$))!
- 7. kérdés: Az $x[4]$ és $x[7]$ elemek összehasonlítása után, melyik két elem kerül összehasonlításra?
- 8. kérdés: Az $x[6]$ és $x[7]$ elemek összehasonlítása után, melyik két elem kerül összehasonlításra?
- 9–10 kérdések: Összesen hány hasonlítás/csere történik a fent megadott 7-elemű tömb esetén?
- 11–14 kérdések: Általánosan, egy szigorúan növekvő/csökkenő hét-elemű sorozat esetén, hány hasonlítás/csere történik?

Kérdőív részei	Tartalmi egységek	Kérdések száma
Előteszt		
1. blokk	Demográfiai adatok	3
2. blokk	Előzetes programozási évek száma	1
3. blokk	Előzetes programozási ismeretek: kiválasztó rendezés	10
4. blokk	Előzetes programozási ismeretek: buborék-rendezés	10
Utóteszt		
5. blokk	Tanulási módszer	1
6. blokk	Shell-sort: sort első három művelet	6
7. blokk	Shell-sort: melyik a következő művelet?	2
8. blokk	Shell-sort: a megadott 7 elemű tömb esetén, hány összehasonlítás, hány csere	2
9. blokk	Shell-sort: szigorúan növekvő/csökkenő 7 elemű tömb esetén, hány összehasonlítás, hány csere	4

11. táblázat: Kérdőív szerkezete (Sematikus és realisztikus algoritmusvizualizációk ismételt megtekintése, illetve ezek kombinálása)

13.1.3. Harmadik kutatás menete

Az előtesztre, az összes résztvevő számára, az egyetem egyik amfiteátrumában került sor, a Socrative alkalmazás felhasználásával.

A tanulási szakaszban, a három csoport külön-külön termekben vett részt. Mindhárom csoport kétszer tekintette meg a Shell-rendezés AlgoRythmics animációját. Az első megtekintés mindenik csoportnál ugyanúgy történt, 0-interaktivitás üzemmódban (a vizualizáció zavartalan megfigyelése). Ami az A-csoport tagjait illeti, számukra a második megtekintés is 0-interaktivitással történt. A B-csoport esetében a második megtekintés $\frac{1}{2}$ -interaktivitással (előre nem ismert időpillanatokban a vizualizáció megállt, és a résztvevőknek meg kellett mondaniuk a következő műveletet), a C-csoport számára pedig 1-

interaktivitással (a résztvevőknek meg kellett adniuk a teljes lépéssorozatra, hogy melyik a következő elempár, és melyik művelet alkalmazandó rá) történt.

Az utóteszt felvezetése tartalmazta, hogy a kérdések az $x[0\dots 6]=\{1, 19, 7, 8, 12, 11, 9\}$ tömbre vonatkoznak, hogy az algoritmus növekvő sorrendbe rendez, 3 és 1 növekmény-értékekkel dolgozik, és hogy művelet alatt hasonlítást vagy cserét értünk.

- 1–3 kérdések: Melyik elempárokra vonatkozik az első/második/ harmadik „Művelet ($x[?], x[?]$)”?
- 4–6 kérdések: Az első/ második/ harmadik művelet: hasonlít vagy csere?
- kérdés: Az $x[3]$ és $x[6]$ elemek összehasonlítása után melyik a következő elempár, amely összehasonlításra kerül?
- kérdés: Az $x[5]$ és $x[6]$ elemek összehasonlítása után melyik a következő elempár, amely összehasonlításra kerül?
- 9–12 kérdések (általánosítás; a növekmény 3 és 1): Bármely szigorúan növekvő/csökkenő hételemű sorozat esetén hány összehasonlítás/ csere történik?
- 13. kérdés: Melyik módszer (interaktivitási szint) tűnik számodra a leghatékonyabbnak (függetlenül attól, hogy melyik csoportba tartoztál)?

Kérdőív részei	Tartalmi egységek	Kérdések száma
Előteszt		
1. blokk	Adatvédelmi irányelvek	1
2. blokk	Demográfiai adatok	3
3. blokk	Előzetes programozási ismeret	5
4. blokk	Logikai feladatok (http://e-hod.elte.hu)	8
Utóteszt		
5. blokk	Tanulási módszer	1
6. blokk	Shell-sort: sort első három művelet	6
7. blokk	Shell-sort: melyik a következő művelet?	2
8. blokk	Shell-sort: szigorúan növekvő/csökkenő 7 elemű tömb esetén, hány összehasonlítás, hány csere	4
9. blokk	Az interaktivitási szintek kiértékelése hatékonyságuk alapján	1

12. táblázat: Kérdőív szerkezete (Az interaktivitás szerepe online algoritmusvizualizációs környezetekben)

13.1.4. Negyedik kutatás menete

Az első szakasz egy amfiteátrumban zajlott mind a 181 résztvevő számára. A résztvevők megkapták a három kérdéssorozatot (papíron), majd mindhárom algoritmus kapcsán a következő lépések zajlottak le: (1) megtekintették (a tanár laptopjáról kivetítve) az algoritmusvizualizációját (a keresések esetén kétszer a tánc-koreográfiát, a rendezés esetén először a tánc-koreográfiát, majd az animációt); (2) válaszoltak a megfelelő kérdéssorozatra (papíron). Az alábbiakban egy $Q_{x.y.z}$ kód úgy értelmezendő, hogy az x szakaszban, az y algoritmus-hoz kapcsolódó z kérdés.

A lineáris keresésére vonatkozó kérdések az alábbiak voltak:

- (Q1.1.1–3) Legszerencsésebb (legjobb) esetben hány táncból találja meg a fiú a párját (amennyiben az illető lány ott van a sorban)? Mi a válaszod, ha
 - 7 lány van?
 - 31 lány van?
 - n lány van?
- (Q1.1.46) A legszerencsétlenebb esetben (legrosszabb) hány táncot feltételez? Mi a válaszod, ha
 - 7 lány van?
 - 31 lány van?
 - n lány van?
- (Q1.1.7) Mi számít a legrosszabb esetnek?

A bináris keresés kapcsán hasonló kérdéseket kellett megválaszolniuk a résztvevőknek (Q1.2.1–7). Azért választottuk a számsorozat hosszának a 7-et, valamint a 31-et, hogy az algoritmus minden lépésében legyen egy jól meghatározott középső elem.

A buborékrendezéshez csatolt kérdések a következők voltak:

- (Q1.3.1) Tegyük fel, hogy a 10 táncost már eleve növekvő sorrendbe állítják fel. De te erről nem tudsz, hiszen bekötötték a szemed. Jogodban áll újra és újra kiválasztani (azaz kitapogatni) két-két táncost, és megkérdezni, hogy melyikük száma nagyobb. Ha a buborék stratégia szerint gondolkozol, akkor hány rákérdezés után tudnád biztosra kijelenteni, hogy a táncosok rendezett sorban vannak?

- (Q1.3.2) Ha a 10 táncos kezdetben csökkenő sorrendbe van rendezve, akkor a buborékstratégia értelmében, hány cserére van szükség a növekvő sorrendbe való átrendezéshez?

A szintéziskérdésre a résztvevők ezt követően kellett, hogy válaszoljanak.

- (Q1.4.1) Mikor célszerű a rendezés+bináris-keresést használni a lineáris-keresés helyett? (a) nagyon hosszú a számsorozat; (b) nagyon sokszor kell újra és újra keresni a számsorozatban; (c) nagyon nagy értékeket tartalmaz a számsorozat; (d) más.

A második szakaszra, amint már említettük, szemináriumi termekben került sor, kérdés-felelet megbeszélés formájában.

A harmadik szakasz is szeminárium termekben zajlott: a résztvevők megkapták a kiválasztó rendezésre vonatkozó kérdéssorozatot, megtekintették az algoritmusnak először a táncokoreográfia ábrázolását, majd az animációját, és végül válaszoltak a kérdésekre. A buborékrendezéshez csatolt kérdéseket kiegészítettük további két kérdéssel. Mivel a második szakasz alatt nyilvánvalóvá vált, hogy elég sok diák számára nem volt magától értetődő, hogy egy rendezési algoritmus legjobb esetben egyetlen cserét sem végez, és hogy legrosszabb esetben minden összehasonlítást csere követ, ezért a kiválasztó rendezés kapcsán ezekre az esetekre is rákérdeztünk.

A kutatás főbb lépéseit a 13. táblázat: A kutatás lépéseit foglalja össze.

Bár, ahogy fentebb utaltunk is rá, a kérdésekre úgy tekintettünk, mint amelyek a tanmenet része, a válaszokat fel tudtuk használni a módszer kiértékelésében is.

		A-csoport (kísérleti)	B-csoport (kontroll-1)	C-csoport (kontroll-2)
Előzetes programozói ismeretek (középiskolából)		Nincs	A két keresési algoritmus és a négyzetes időbonyolultságú rendezési algoritmusok alapfokon (algoritmus bonyolultsági fogalmak nélkül)	A két keresési algoritmus és több rendezési algoritmus, valamint az ezekkel kapcsolatos bonyolultsági fogalmak
	Első szakasz		Lineáris keresés: tánc-koreográfia 2-szer 7 algoritmus bonyolultsági kérdés (Q1.1.1-7) Bináris keresés: tánc-koreográfia 2-szer 7 algoritmus bonyolultsági kérdés (Q1.2.1-7) Buborékredezés: tánc-koreográfia + animáció 2 algoritmus bonyolultsági kérdés (Q1.3.1-2) 1 szintézis kérdés (Q1.4.1)	
Tanulási-tesztelési folyamat (mielőtt elkezdődött volna a Programozás-I kurzus)	Második szakasz	A tanár kérdések segítségével rávezetni a tanulókat az első szakaszbeli helyes válaszokra	–	–
	Harmadik szakasz	Kiválasztó rendezés: tánc-koreográfia + animáció 4 algoritmus bonyolultsági kérdés (Q3.1.1-4)	–	–

13. táblázat: A kutatás lépései

	Lineáris keresés (lineáris komplexitás) Q1.1.1-6	Bináris keresés (logaritmikus komplexitás) Q1.2.1-5	Buborékredezés (négyzetes komplexitás) Q1.3.1-2	Szintézis kérdés Q1.4.1
A-csoport	79%	34%	59%	51%
B-csoport	86%	53%	49%	52%
C-csoport	94%	76%	83%	65%

14. táblázat: Az A, B, C csoportok első szakaszbeli eredményei (mivel a Q1.2.6 kérdés megválaszolása viszonylag magas matematikai ismeretet igényelt, ezt kihagytuk elemzésünkéből)

13.1.5. Ötödik kutatás menete

Mivel az alkalmazott módszer ötvözte az informatikát a tánccal, ezért a résztvevők elsőként a következő kérdésekre kellett, hogy válaszoljanak:

- K_{0,1} kérdés: Mennyire szereted a számítógépezést és az informatikát? (1-es azt jelenti, hogy nagyon nem szereted, 10-es azt jelenti, hogy nagyon szereted)
- K_{0,2} kérdés: Mennyire szereted a táncot és a zenét? (1-es azt jelenti, hogy nagyon nem szereted, 10-es azt jelenti, hogy nagyon szereted)

Ezt követte a vizsgálat első szakasza: a párját kereső fiú történetét egy kép segítségével szemléltettük. A kép egy fiút ábrázolt, aki a mellén egy számot viselt. A vele szemben ülő lányok, a hátukon viselték a számokat, és egyik közülük a fiú párja volt (30. ábra). A történetbeli fiú feladata az volt, hogy felkérve a lányokat egy-egy táncra, találja meg a párját, vagyis azt a lányt, aki ugyanazt a számot viseli, és ugyanazt a tánckoreográfiát táncolja. A tanulóknak azonosulniuk kellett a táncos fiúval, majd a kép tanulmányozása után a tanulók az alábbi kérdésre kellett választ adjanak:

- K₁ kérdés: Hány lánnyal kell táncolnia a fiúnak ahhoz, hogy megtalálja a párját?



30. ábra: A párját kereső fiú és a hételemű lánysorozat

Ezt követte a második szakasz, amely a lineáris keresést ábrázoló videó megtekintéséből állt (31. ábra). Ezután felkértük a résztvevőket, hogy válaszoljanak arra a kérdésre, hogy:

- K₂ kérdés: Mennyire értetted meg a videón látott fiú alkalmazta keresési stratégiát? (1: egyáltalán nem; 10: teljes mértékben)



31. ábra: Lineáris keresés, videó

A harmadik szakasz 5 kérdés köré épült. Hangsúlyoztuk a résztvevőknek, hogy ezeket a kérdéseket a videón látott fiú alkalmazta módszer szerint kell megválaszolniuk. Az 1. kérdés egy hasonló képre épült, mint amelyet az első

szakaszban láttak a résztvevők, csak hogy a fiúnak 10 lány között kellett keresnie (32. ábra). A csatolt kérdés is hasonló volt:

- $K_{3.1}$ kérdés: Hány lánnyal kell táncolnia a fiúnak, ahhoz, hogy megtalálja a párját?



32. ábra: Lineáris keresés, tizelemű sorozaton

Ahhoz hogy felmérjük, hogy a résztvevők mennyire mélyen értették meg a vizualizált algoritmust, a következő algoritmus bonyolultsági szempontokra kérdeztünk rá. A következő kérdéspár ($K_{3.2.1}$ és $K_{3.2.2}$ kérdések) a 33. ábrahoz kapcsolódott:



33. ábra: Lineáris keresés, best/worst case, 5 elemű számsorozaton

- K_{3.2.1} kérdés: Milyen számokkal látnád el a lányokat, ahhoz, hogy a fiú egy tánc után megtalálja a párját? („legszerencsésebb eset”). Sorold fel az 5 számot, szóközzel elválasztva.
- K_{3.2.2} kérdés: Milyen számokkal látnád el a lányokat, ahhoz, hogy a fiú a „legszerencsétlenebb eset”-ben legyen? Sorold fel az 5 számot, szóközzel elválasztva.

Az utolsó kérdés-páros (K_{3.3.1} és K_{3.3.2} kérdések) a 34. ábraán alapult, és azt tesztelte, hogy a résztvevők képesek-e általánosítani az előző kérdésekre adott válaszaikat:

- K_{3.3.1} kérdés: Hány lánnyal kell táncoljon a fiú „legszerencsésebb esetben” ahhoz, hogy megtalálja a párját, ha 5 lány helyett, x lány állna vele szemben?

- K_{3.3.2} kérdés: Hány lánnyal kell táncoljon a fiú „legszerencsétlenebb esetben” ahhoz, hogy megtalálja a párját, ha 5 lány helyett, x lány állna vele szemben?



34. ábra: Lineáris keresés, best case/worst case, ismeretlen számú számsorozaton

Mindezt követően, egy osztálytermi megbeszélés keretében, a résztvevőket felkértük, hogy indokolják meg válaszaikat.

Kérdőív részei	Tartalmi egységek	Kérdések száma
Első szakasz		
1. blokk	Demográfiai adatok	1
2. blokk	Informatika/tánc iránti szere- tet	2
3. blokk	Számítógépes gondolkodá- suk szintjének mérése	1
4. Harmadik szakasz		
5. blokk	Algoritmus megértése	1
6. blokk	Bonyolultsági kérdések 5 elemű sorozatra	2
7. blokk	Bonyolultsági kérdések x hosszú sorozatra	2

15. táblázat: Kérdőív szerkezete (Számítógépes gondolkodás vizsgálata különböző korosztályú tanulók-
nál)

14. FÜGGELÉK

14.1. A függelék: Számítógépes gondolkodás teszt

9. Alex és Bea titkosított üzeneteket küldenek egymásnak. Ehhez minden szót külön kódolnak, mégpedig három lépésben:

1. lépés: megfordítják a szót (a betűk sorrendjét);
2. lépés: minden betűt két hellyel **balra** tolnak (körkörösén, tehát az első betű az utolsó előtti helyre kerül, stb.);
3. lépés: minden magánhangzót (azaz: A, E, I, O, U) kicserélnek a rákövetkező magánhangzóra (az utolsó magánhangzó, az U az elsőre, az A-ra változik).

Példa: BEAR >> RAEB >> EBRA >> IBRE

KÉRDÉS: Mi lesz az **ELSŐ** lépés után az üzenet, ha kezdetben **SUMMER**?
(Válaszodat add meg CSUPA NAGYBETŰVEL az angol ABC betűit használva)

10. UGYANAZ A TITKOSÍTÁSI MÓDSZER (Alex és Bea titkosított üzeneteket küldenek egymásnak).

Ehhez minden

szót külön kódolnak, mégpedig három lépésben:

1. lépés: megfordítják a szót (a betűk sorrendjét);
2. lépés: minden betűt két hellyel **balra** tolnak (körkörösén, tehát az első betű az utolsó előtti helyre kerül, stb.);
3. lépés: minden magánhangzót (azaz: A, E, I, O, U) kicserélnek a rákövetkező magánhangzóra (az utolsó magánhangzó, az U az elsőre, az A-ra változik).

Példa: BEAR >> RAEB >> EBRA >> IBRE)

KÉRDÉS: Mi lesz az **MÁSODIK** lépés után az üzenet, ha kezdetben **SAPIENTIA**? (Válaszodat add meg CSUPA NAGYBETŰVEL az angol ABC betűit használva)

11. UGYANAZ A TITKOSÍTÁSI MÓDSZER: (Alex és Bea titkosított üzeneteket küldenek egymásnak.

Ehhez minden

szót külön kódolnak, mégpedig három lépésben:

1. lépés: megfordítják a szót (a betűk sorrendjét);
2. lépés: minden betűt két hellyel **balra** tolnak (körkörösén, tehát az első betű az utolsó előtti helyre kerül, stb.);
3. lépés: minden magánhangzót (azaz: A, E, I, O, U) kicserélnék a rákövetkező magánhangzóra (az utolsó magánhangzó, az U az elsőre, az A-ra változik).

Példa: BEAR >> RAEB >> EBRA >> IBRE)

KÉRDÉS: Mi lesz a titkosított üzenet a **HARMADIK** lépés után, ha kezdetben **INHERITANCE**?
(Válaszodat add meg CSUPA NAGYBETŰVEL az angol ABC betűit használva)

12. UGYANAZ A TITKOSÍTÁSI MÓDSZER: (Alex és Bea titkosított üzeneteket küldenek egymásnak.

Ehhez minden

szót külön kódolnak, mégpedig három lépésben:

1. lépés: megfordítják a szót (a betűk sorrendjét);
2. lépés: minden betűt két hellyel **balra** tolnak (körkörösén, tehát az első betű az utolsó előtti helyre kerül, stb.);
3. lépés: minden magánhangzót (azaz: A, E, I, O, U) kicserélnék a rákövetkező magánhangzóra (az utolsó magánhangzó, az U az elsőre, az A-ra változik).

Példa: BEAR >> RAEB >> EBRA >> IBRE)

KÉRDÉS: Mi volt az **eredeti** szó, ha a titkosított üzenet **NRAUJYI**?
(Válaszodat add meg CSUPA NAGYBETŰVEL az angol ABC betűit használva)

13. A hódok és a viperák között rendezett úszóversenyen tizenketten vettek részt. A következő pontokat kapták: 2, 2, 2, 4, 4, 4, 4, 4, 4, 4, 4, 4.

Sajnos a viperák nem voltak túl sikeresek: egyik viperának sem volt több, vagy ugyanannyi pontja, mint bármelyik hódnak.

Hány **hód** vett részt a versenyen?

- ☐ A 3
- ☐ B 6
- ☐ C 9
- ☐ D 12

- 14.** A hódok és a viperák között rendezett úszóversenyen tízen vettek részt. A következő pontokat kapták: 1, 2, 3, 4, 5, 5, 6, 7, 8, 9.

Sajnos a viperák nem voltak túl sikeresek: egyik viperának sem volt több pontja, mint bármelyik hódnak. De egy viperának épp annyi pontja volt, mint egy hódnak.

Melyik versenyzőből van több résztvevő?

- ☐ A Hódból van több résztvevő
- ☐ B Viperából van több résztvevő
- ☐ C A hód és vipera résztvevők száma egyforma

- 15.** A hódok és a viperák között rendezett úszóversenyen kilencen vettek részt. A következő pontokat kapták: 1, 2, 2, 3, 4, 5, 5, 6, 7.

Sajnos a viperák nem voltak túl sikeresek: egyik viperának sem volt több pontja, mint bármelyik hódnak. De egy viperának épp annyi pontja volt, mint egy hódnak és ezenkívül két vipera is egyforma pontszámot ért el.

Hány **vipera** vett részt a versenyen?

- ☐ A 2
- ☐ B 6
- ☐ C 7
- ☐ D 9

- 16.** A hódok és a viperák között rendezett úszóversenyen tizenketten vettek részt. A következő pontokat kapták: 3, 1, 6, 3, 8, 10, 4, 5, 9, 4, 7, 2.

Sajnos a viperák nem voltak túl sikeresek: egyik viperának sem volt több pontja, mint bármelyik hódnak. De egy viperának épp annyi pontja volt, mint egy hódnak és ezenkívül két hód is egyforma pontszámot ért el.

Hány **hód** vett részt a versenyen?

- ☐ A 3
- ☐ B 5
- ☐ C 7
- ☐ D 9

14.2. B függelék: A buborék- és kiválasztó rendezések pszeudókódja

Buborékredezés kódja az *animáció* alapján

```
int i = n-1;
do {
    last_swap_index = 0;
    for(int j = 0; j <= i-1; ++j) {
        if(a[j] > a[j+1]) {
            swap(a[j], a[j+1]);
            last_swap_index = j;
        }
    }
    i = last_swap_index;
} while (last_swap_index > 0);
```

Buborékredezés kódja a *tánc Koreográfia* alapján

```
int i = n-1;
do
{
    last_swap_index = 0;
    for(int j = 0; j <= i-1; ++j)
    {
        if(a[j] > a[j+1])
        {
            swap(a[j], a[j+1]);
            last_swap_index = j;
        }
    }
    i = last_swap_index;
}
while (last_swap_index > 0);
```

Kiválasztó rendezés kódja az *animáció* alapján

```
for(int i = 0; i < n-1; ++i) {
    min_idx = i;
    for(int j = i+1; j < n; ++j) {
        if(a[j] < a[min_idx]) {
            min_idx = j;
        }
    }
    swap(a[i], a[min_idx]);
}
```

Kiválasztó rendezés kódja a *tánc Koreográfia* alapján

```
for(int i = 0; i < n-1; ++i)
{
    for(int j = i+1; j < n; ++j)
    {
        if(a[j] < a[i])
        {
            swap(a[j], a[i]);
        }
    }
}
```

15. ÁBRAJEGYZÉK

1. ábra: Rekurzív függvény „eljátszva”.....	18
2. ábra: Buborékrendezés	19
3. ábra: Rendezési algoritmusok videói	20
4. ábra: Új algoritmusok 2016-tól	22
5. ábra: AlgoRhythmics új dimenziói	23
6. ábra: Rendezési algoritmusok	24
7. ábra: Tanulási lépések listája.....	24
8. ábra: Kiértékelő tesztek.....	26
9. ábra: Profil oldal.....	26
10. ábra: A táncról a kódig	27
11. ábra: A két csoport előteszt és utóteszt eredményei.....	38
12. ábra: Buborék vs. Kiválasztó rendezés eredményei az utóteszten	40
13. ábra: Statikus vs. dinamikus ábrázolásmód.....	48
14. ábra: A csoportonkénti eredmények átlagainak ábrázolása	51
15. ábra: Előteszt és utóteszt eredményei csoportonként.....	61
16. ábra: A 3 programozási szinten levő diákok eredményei az utóteszt során, csoportonként	62
17. ábra: Az előteszt és utóteszt eredményei összesítve nemenként.....	63
18. ábra: Az előteszt és utóteszt eredményei nemenként csoportokra lebontva	63
19. ábra: Az 1-6., 7-8., 9-12. kérdések válaszai az 1 vagy 2 évet tanultak ese- tén.....	64
20. ábra: A diákok véleménye a leghatékonyabb tanulási módszerről	65
21. ábra: Lineáris keresés flamenco tánccal.....	71
22. ábra: Bináris keresés flamenco tánccal	71

23. ábra: Az kísérlet első szakaszának eredményei korosztály szerint (mindkét iskola)	88
24. ábra: A kísérlet harmadik szakasz eredményei korosztály szerint (mindkét iskola)	88
25. ábra: A kísérlet harmadik szakaszának eredményei, az utolsó két kérdés kivételével, korosztály szerint (mindkét iskola).....	90
26. ábra: A kísérlet első szakaszának eredményeinek feldolgozása iskolák szerint	93
27. ábra: A kutatási kérdésekre adott válaszok feldolgozása nemenként	96
28. ábra: A Shell-rendezés realiztikus ábrázolása	115
29. ábra: A Shell-rendezés sematikus ábrázolása	115
30. ábra: A párját kereső fiú és a hételemű lánysorozat.....	124
31. ábra: Lineáris keresés, videó	124
32. ábra: Lineáris keresés, tízelemű sorozaton.....	125
33. ábra: Lineáris keresés, best/worst case, 5 elemű számsorozaton	126
34. ábra: Lineáris keresés, best case/worst case, ismeretlen számú számsorozaton	127

16. TÁBLÁZATJEGYZÉK

1. táblázat: Fisher-féle egzakt tesztjének eredménye	36
2. táblázat: Elő- és utóteszt eredmények a két csoport szignifikancia különbségének mérésére.....	38
3. táblázat: A csoportonkénti eredmények átlagai és zárójelben a megfelelő szórás értékek	50
4. táblázat: Résztvevők átlagai csoportokra bontva	60
5. táblázat: Az A csoport első és második szakaszbeli eredményei	76
6. táblázat: A kutatásban résztvevők száma korosztályonként, nemenként, iskolánként	84
7. táblázat: A kutatás kérdéseire adott válaszok átlaga és zárójelben a megfelelő szórás érték	86
8. táblázat: Az első kérdésre adott helyes és helytelen válaszok száma osztályonként	87
9. táblázat: Adott iskolában tanult évek száma szakterületenként	91
10. táblázat: Kérdőív szerkezete (Sematikus versus emberi mozgást alkalmazó realisztikus algoritmusvizualizáció)	113
11. táblázat: Kérdőív szerkezete (Sematikus és realisztikus algoritmusvizualizációk ismételt megtekintése, illetve ezek kombinálása)	117
12. táblázat: Kérdőív szerkezete (Az interaktivitás szerepe online algoritmusvizualizációs környezetekben)	119
13. táblázat: A kutatás lépései	122
14. táblázat: Az A, B, C csoportok első szakaszbeli eredményei (mivel a Q1.2.6 kérdés megválaszolása viszonylag magas matematikai ismeretet igényelt, ezt kihagytuk elemzésünkéből)	122
15. táblázat: Kérdőív szerkezete (Számítógépes gondolkodás vizsgálata különböző korosztályú tanulóknál)	128

17. IRODALOMJEGYZÉK

- Ahadi, A., Lister, R., Lal, S., Leinonen, J., & Hellas, A. (2017, January). Performance and Consistency in Learning to Program. In *Proceedings of the Nineteenth Australasian Computing Education Conference* (pp. 11-16). ACM.
- Aho, A. V. (2011). Computation and computational thinking. *Ubiquity Symposium*. DOI: 10.1145/1895419.1922682
- Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, 55(7), 832–835.
- Ainsworth, S., & VanLabeke, N. (2004). Multiple forms of dynamic representation. *Learning and instruction*, 14(3), 241-255.
- AlgoRythmicsenvironment. (2020). <https://algorhythmics.ms.sapientia.ro/>.
- Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75, 661-670.
- Ayres, P., & Paas, F. (2007). Making instructional animations more effective: A cognitive load approach. *Applied Cognitive Psychology*, 21, 695–700.
- Berney, S., & Betrancourt, M. (2016). Does animation enhance learning? A meta-analysis. *Computers & Education*, 101, 150–167.
- Butcher, K. R. (2006). Learning from text with diagrams: Promoting mental model development and inference generation. *Journal of Educational Psychology*, 98(1), 182.
- Byrne, M. D., Catrambone, R., & Stasko, J. T. (1996). Do algorithm animations aid learning?. Georgia Institute of Technology.
- Byrne, P., & Lyons, G. (2001, June). The effect of student attributes on success in programming. In *ACM SIGCSE Bulletin* (Vol. 33, No. 3, pp. 49-52). ACM.
- Carney, R. N., & Levin, J. R. (2002). Pictorial illustrations still improve students' learning from text. *Educational psychology review*, 14(1), 5-26.
- Castro-Alonso, J. C., Ayres, P., & Paas, F. (2014). Dynamic visualisations and motor skills. In *Handbook of human centric visualization* (pp. 551-580). Springer, New York, NY.
- Castro-Alonso, J. C., Ayres, P., Wong, M., & Paas, F. (2018). Learning symbols from permanent and transient visual presentations: Don't overlay the hand. *Computers & Education*, 116, 1–13.
- Comenius (1992): *Pampedia*. (IV.rész, ford. Bollók János), B-A-Z Levéltár, Sárospatak. IX. fejezet, A kisgyermekkor iskolája, az anyaöl, vagyis az emberi sarjnak a születéstől körülbelül hatéves koráig tartó gondos nevelése. 68-80.
- de Koning, B. B., & Tabbers, H. K. (2011). Facilitating understanding of movements in dynamic visualizations: An embodied perspective. *Educational Psychology Review*, 23(4), 501–521.
- Denning, P. J. (2009). The profession of IT Beyond computational thinking. *Communications of the ACM*, 52(6), 28-30.

- Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, 60(6), 33-39.
- Evans, G. E., & Simkin, M. G. (1989). What best predicts computer proficiency?. *Communications of the ACM*, 32(11), 1322-1327.
- Feaster, Y., Ali, F., Zhai, J., & Hallstrom, J. O. (2014, June). Serious toys: three years of teaching computer science concepts in K-12 classrooms. In *Proceedings of the 2014 conference on Innovation & technology in computer science education* (pp. 69-74). ACM.
- Fritz, C. O., Morris, P. E., Bjork, R. A., Gelman, R., & Wickens, T. D. (2000). When further learning fails: Stability and change following repeated presentation of text. *British Journal of Psychology*, 91(4), 493-511.
- Goldstone, R. L., & Son, J. Y. (2005). The transfer of scientific principles using concrete and idealized simulations. *The Journal of the Learning Sciences*, 14(1), 69-110.
- Graesser, A. C., & Person, N. K. (1994). Question asking during tutoring. *American educational research journal*, 31(1), 104-137.
- Guzdial, M., & Tew, A. E. (2006, September). Imagineering inauthentic legitimate peripheral participation: an instructional design approach for motivating computing education. In *Proceedings of the second international workshop on Computing education research* (pp. 51-58). ACM.
- Höffler, T. N., & Leutner, D. (2007). Instructional animation versus static pictures: A meta-analysis. *Learning and Instruction*, 17(6), 722-738.
- Höffler, T. N., Leutner, D. (2007). Instructional animation versus static pictures: A meta-analysis. *Learning and instruction*, 17(6), 722-738.
- Jarc, D. J., Feldman, M. B., & Heller, R. S. (2000). Assessing the benefits of interactive prediction using web-based algorithm animation courseware. *ACM SIGCSE Bulletin*, 32(1), 377-381.
- Katai, Z. (2014, June). Selective hiding for improved algorithmic visualization. In *Proceedings of the 2014 conference on Innovation & technology in computer science education* (pp. 33-38). ACM.
- Káta, Z. (2015). The challenge of promoting algorithmic thinking of both sciences-and humanities-oriented learners. *Journal of Computer Assisted Learning*, 31(4), 287-299.
- Katai, Z., Toth, L., & Adorjani, A. K. (2014). Multi-Sensory Informatics Education. *Informatics in Education*, 13(2), 225-240.
- Káta, Z., Juhász, K., & Adorjáni, A. K. (2008). On the role of senses in education. *Computers & Education*, 51(4), 1707-1717.
- Korhonen, A., & Malmi, L. (2000). Algorithm simulation with automatic assessment. *ACM SIGCSE Bulletin*, 32(3), 160-163.
- Lau, W. W., & Yuen, A. H. (2009). Exploring the effects of gender and learning styles on computer programming performance: implications for programming pedagogy. *British Journal of Educational Technology*, 40(4), 696-712.
- Lewandowski, G., Bouvier, D. J., Chen, T. Y., McCartney, R., Sanders, K., Simon, B., & VanDeGrift, T. (2010). Commonsense understanding of concurrency: computing students and concert tickets. *Communications of the ACM*, 53(7), 60-70.

- Lin, L., & Li, M. (2018). Optimizing learning from animation: Examining the impact of bio-feedback. *Learning and Instruction*, 55, 32–40.
- Lowe, R. K. (1999). Extracting information from an animation during complex visual learning. *European Journal of Psychology of Education*, 14, 225–244.
- Lowe, R. K., & Boucheix, J.-M. (2017). A composition approach to design of educational animations. In R. Lowe, & R. Ploetzner (Eds.). *Learning from dynamic Visualizations: Innovations in research and application* (pp. 5–30). Berlin: Springer.
- Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014, June). 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.
- Mayer, R. E. (2005). Cognitive theory of multimedia learning. In R. E. Mayer (Ed), *The Cambridge handbook of multimedia learning* (pp. 31-48). Cambridge University Press.
- Mayer, R. E., & Moreno, R. (2003). Nine ways to reduce cognitive load in multimedia learning. *Educational psychologist*, 38(1), 43-52.
- McCartney, R., Bouvier, D. J., Chen, T. Y., Lewandowski, G., Sanders, K., Simon, B., & VanDeGrift, T. (2009, August). Commonsense computing (episode 5): algorithm efficiency and balloon testing. In Proceedings of the fifth international workshop on Computing education research workshop (pp. 51-62). ACM.
- Moreno, R., Ozogul, G., & Reisslein, M. (2011). Teaching with concrete and abstract visual representations: Effects on students' problem solving, problem representations, and learning perceptions. *Journal of Educational Psychology*, 103, 32–47.
- Naps, T. L., Eagan, J. R., & Norton, L. L. (2000, May). JHAVÉ—an environment to actively engage students in Web-based algorithm visualizations. In ACM SIGCSE Bulletin (Vol. 32, No. 1, pp. 109-113). ACM.
- Nugteren, M. L., Tabbers, H. K., Scheiter, K., & Paas, F. (2014). Simultaneous and Sequential Presentation of Realistic and Schematic Instructional Dynamic Visualizations. In *Handbook of human centric visualization* (pp. 605-622). Springer, New York, NY.
- Paas, F., & Sweller, J. (2012). An evolutionary upgrade of cognitive load theory: Using the human motor system and collaboration to support the learning of complex cognitive tasks. *Educational Psychology Review*, 24(1), 27-45.
- Paas, F., & Sweller, J. (2012). An evolutionary upgrade of cognitive load theory: Using the human motor system and collaboration to support the learning of complex cognitive tasks. *Educational Psychology Review*, 24(1), 27-45.
- Pukánszky Béla, N. (n.d.). Neveléstörténet. Bp., Nemzeti Tankvk., 1997. 185. o
- Rau, M. A. (2013). *Conceptual learning with multiple graphical representations: Intelligent tutoring systems support for sense-making and fluency-building processes* (Unpublished doctoral dissertation). Carnegie Mellon University, Pittsburgh, PA.
- Rau, M. A. (2017). Conditions for the effectiveness of multiple visual representations in enhancing STEM learning. *Educational Psychology Review*, 29(4), 717-761.
- Rau, M. A., & Matthews, P. G. (2017). How to make 'more'better? Principles for effective use of multiple representations to enhance students' learning about fractions. *ZDM*, 49(4), 531-544.

- Rau, M. A., Aleven, V., & Rummel, N. (2013). Interleaved practice in multi-dimensional learning tasks: Which dimension should we interleave?. *Learning and Instruction*, 23, 98-114.
- Rau, M. A., Michaelis, J. E., & Fay, N. (2015). Connection making between multiple graphical representations: A multi-methods approach for domain-specific grounding of an intelligent tutoring system for chemistry. *Computers & Education*, 82, 460-485.
- Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, 72, 678-691.
- Román-González, M., Pérez-González, J. C., Moreno-León, J., & Robles, G. (2018). Can computational talent be detected? Predictive validity of the Computational Thinking Test. *International Journal of Child-Computer Interaction*, 18, 47-58.
- Roy, M., & Chi, M. T. (2005). The self-explanation principle in multimedia learning. *The Cambridge handbook of multimedia learning*, 271-286.
- Scheiter, K., Gerjets, P., Huk, T., Imhof, B., & Kammerer, Y. (2009). The effects of realism in learning with dynamic visualizations. *Learning and Instruction*, 19(6), 481-494.
- Settle, A., Goldberg, D. S., & Barr, V. (2013, July). Beyond computer science: computational thinking across disciplines. In *Proceedings of the 18th ACM conference on Innovation and technology in computer science education* (pp. 311-312). ACM.
- Shaffer, C. A., Cooper, M. L., Alon, A. J. D., Akbar, M., Stewart, M., Ponce, S., & Edwards, S. H. (2010). Algorithm visualization: The state of the field. *ACM Transactions on Computing Education (TOCE)*, 10(3), 9.
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142-158.
- Simon, B., Chen, T. Y., Lewandowski, G., McCartney, R., & Sanders, K. (2006, September). Commonsense computing: what students know before we teach (episode 1: sorting). In *Proceedings of the second international workshop on Computing education research* (pp. 29-40). ACM.
- Simon, B., Kinnunen, P., Porter, L., & Zazkis, D. (2010, June). Experience report: CS1 for majors with media computation. In *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education* (pp. 214-218). ACM.
- Stout, J., & Tamer, B. (2016, February). Collaborative learning eliminates the negative impact of gender stereotypes on women's self-concept. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (pp. 496-496). ACM.
- Sweller, J., Van Merriënboer, J. J., & Paas, F. G. (1998). Cognitive architecture and instructional design. *Educational psychology review*, 10(3), 251-296.
- Sweller, J., Clark, R., & Kirschner, P. (2010). Teaching general problem-solving skills is not a substitute for, or a viable addition to, teaching mathematics. *Notices of the American Mathematical Society*, 57(10), 1303-1304.
- Tew, A. E., McCracken, W. M., & Guzdial, M. (2005, October). Impact of alternative introductory courses on programming concept understanding. In *Proceedings of the first international workshop on Computing education research* (pp. 25-35). ACM.

- Tofade, T., Elsner, J., & Haines, S. T. (2013). Best practice strategies for effective use of questions as a teaching tool. *American journal of pharmaceutical education*, 77(7).
- Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59(236), 433.
- Tversky, B., Morrison, J. B., & Betrancourt, M. (2002). Animation: can it facilitate?.
- Werth, L. H. (1986). Predicting student performance in a beginning computer science class (Vol. 18, No. 1, pp. 138-143). ACM.
- Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717-3725.

18. FORRÁSJEGYZÉK

AlgoRythmicschannel. (2011). <https://www.youtube.com/user/AlgoRythmics>.

Bebras.(2020).International Challenge on Informaticsand Computational Thinking<https://www.bebbras.org/>.

CSTA. (2017). CSTA K-12 Computer Science Standards, Revised 2017. Retrieved from: <https://www.csteachers.org/page/standards>.

European Schoolnet. (2015). Computing our future. Computer programming and coding: priorities, school and initiatives across Europe [Technical report]. Retrieved from: <http://www.eun.org/resources/detail?publicationID=661>.
<http://www2.edu.ro/index.php/programe>

New Media. (2013). *Shell-sort*. Retrieved 01.01.19 from. <https://www.youtube.com/watch?v=M9YCh-ZeC7Y>.

19. PUBLIKÁCIÓS JEGYZÉK

Idegen nyelvű tudományos konferencia közlemények

[Q2] Kátai, Z., & Osztián, E. (2021). Improving AlgoRythmics Teaching-Learning Environment by Asking Questions. *International Journal of Instruction*, 14(2).

Osztián, Pálma Rozalia, Zoltán Kátai, and Erika Osztián. "Algorithm Visualization Environments: Degree of interactivity as an influence on student-learning." In 2020 IEEE Frontiers in Education Conference (FIE), pp. 1-8. IEEE, 2020.

Kátai Zoltán, Osztián Erika, Osztián Pálma Rozália, Nagy Eszter Jáhel, Cosma Cristian, Looking for the Optimal Interactivity Level in the AlgoRythmics Learning Environment. In: Theo Bastiaens (szerk.): *Proceedings of EdMedia + Innovate Learning*, Amsterdam, Association for the Advancement of Computing in Education, 2019, pp. 106–114. "Looking for the Optimal Interactivity Level in the AlgoRythmics Learning Environment - Learning & Technology Library (LearnTechLib)" n.d.

[Q4] Kátai, Z. Osztián, E., Lőrincz, B.: The effect of multiple graphical representations on learning algorithms In: *The online Journal of Educational Technology International Conference on New Horizons in Education*, Paris, [814,819], 2018

[Q4] Osztián, E., Kátai, Z., Vekov, G.: Multi dimensional expansion of algorithmics In: *INTE International Conference on New Horizons in Education*, IDEC International Distance Education Conference, ITICAL International Trends and Issues in Communication & Media Conference Proceedings Book Vol. 2 July 17-19, 2017, Berlin, Germany / Eds. Aytekin İşman, Ahmet Eskicumali, ASET, Sakarya, 701-706, 2017.

Kátai, Z., Osztián, E., Vekov, G.: Promoting computational thinking by artistically enhanced algorithm visualization In: INFODIDACT 2016 : 9. Informatika Szakmódszertani Konferencia 2016.11.24 2016.11.26, Zamárdi : [elektronikus dokumentum], Webdidaktika Alapítvány, Budapest, [1-12.], 2016

Magyar nyelvű tudományos konferencia közlemények

Kátai Zoltán, Osztián Erika, Algoritmus Vizualizációs Környezetek: Az interaktivitás hatása a tanulási eredményekre. In: Zsakó László, Szlávi Péter (szerk.): INFODIDACT - 2020, Zamárdi, Webdidaktika Alapítvány, 2020, pp. 1–10., ISBN 978-615-80608-4-4

Kátai Zoltán, Osztián Erika, Az algoritmikus gondolkodás vizsgálata különböző korosztályú tanulóknál. In: Zsakó László, Szlávi Péter (szerk.): INFODIDACT - 2019, Zamárdi, Webdidaktika Alapítvány, 2019, pp. 1–10., ISBN 978-615-80608-2-0

Kátai, Z., Osztián, E.: Algoritmikus gondolkodás fejlesztése különböző korosztályú és szakos tanulóknál In: Számítástechnika és Oktatás Konferencia SzámOkt, Tusnádfürdő, [225-230.], 2018

Osztián, E., Kátai, Z., Vekov, G.: Számítógépes gondolkodás fejlesztése egyes oktatási In: 23th Multimedia in Education Conference Proceedings, Presa Universitară Clujeană, Cluj Napoca, [107-112], 2017.

Osztián, P., Kátai, Z., Osztián, E., Vekov, G.: Algoritmika és művészet, avagy kupacrendezés kalotaszegi módra = Algorithms and art Heapsort with traditional Transylvanian dance In: ENELKO 2017 XVIII. Nemzetközi Ener-

getika Elektrotechnika Konferencia, SzámOkt 2017 XXVII. Nemzetközi Számítástechnika és Oktatás Konferencia, Kolozsvár, 2017. október 12 15 / szerk . Biró Károly Ágoston, Sebestyén Pál György, Erdélyi Magyar Műszaki Tudományos Társaság, Cluj Napoca, [1-5], 2017

Kátai , Z., Osztián, E., Vekov , G., Osztián, P.: Algo Ritmika: új dimenziók. In: INFODIDACT 2017 : 10. Informatika Szakmódszertani Konferencia 2017.11.23 2017.11.25., Zamárdi : [elektronikus dokumentum], Webdidaktika Alapítvány, Budapest, [1-11.], 2017

Szakmai konferenciák, előadástartások idegen nyelven

Időpont	Előadás adatai
2020. november	INFODIDACT 2020 Informatika Szakmódszertani Konferencia, Zamárdi (online) Algoritmus Vizualizációs Környezetek: Az interaktivitás hatása a tanulási eredményekre
2020. november	Matematika és Informatika Alkalmazásokkal Konferencia, Kolozsvár (online) Algoritmus Vizualizációs Környezetek: Az interaktivitás hatása a tanulási eredményekre
2020. október	FIE Frontiers in education, Uppsala, Sweden (online) Algorithm Visualization Environments: Degree of interactivity as an influence on student-learning

Publikációs jegyzék

2019. szeptember	International Conference on Mathematics and Informatics-Sapientia, Marosvásárhely Testing tool for investigating young school students' algorithmic thinking
------------------	---

Szakmai konferenciák, előadástartások magyar nyelven

Időpont	Előadás adatai
2019. november	INFODIDACT 2019 Informatika Szakmódszertani Konferencia, Zamárdi Az algoritmikus gondolkodás vizsgálata különböző korosztályú tanulóknál, E-learning tesztelési környezetben
2018. október	Számítástechnika és Oktatás Konferencia – SzámOkt Algoritmikus gondolkodás fejlesztése különböző korosztályú és szakos tanulóknál
2018. július	INTE 2018 International Conference on New Horizons in Education Paris The effect of multiple graphical representations on learning algorithms
2017. november	INFODIDACT 2017 Informatika Szakmódszertani Konferencia, Zamárdi Algo-Ritmika: új dimenziók
2017. október	SzámOkt 2017, Nagyvárad XXVII. Számítástechnika és Oktatás Konferencia 2017

Publikációs jegyzék

Időpont	Előadás adatai
	Algoritmika és művészet, avagy kupacrendezés kalotaszegi módra
2017. július	INTE 2017 International Conference on New Horizons in Education – Berlin, Germany Multi-dimensional expansion of algorithemics
2017. június	XXIII. Multimédia az oktatásban konferencia - Kolozsvár Keresési algoritmusok flamenco stílusban
2016. november	INFODIDACT 2016 konferencia- Zamárdi, Informatika Szakmódszertani Konferencia Számítógépes gondolkodás három tánclépésben
2016. november	A Magyar Tudomány Napja Erdélyben 7. Matematika és Informatika Alkalmazásokkal Konferencia-Kolozsvár Számítógépes gondolkodás három tánclépésben

Résztétel kutatási projekteken

Időtartam	Projekt neve
2019–2020	Collegium Talentum program 2019–2020-as pályázat (Dr. Kátai Zoltán, Osztian Erika)
2017–2018	“Egyetemi Kutatási Program 2017/2018” pályázat (Dr. Kátai Zoltán, Osztian Erika, Osztian Palma I éves hallgató)

Szakmaspecifikus produktumok

Időtartam	Produktum neve
2017–2018	Heap-sort-rendezési algoritmus néptánccal, a Sapientia Erdélyi Magyar Tudományegyetem I-II éves informatika szakos egyetemisták közreműködésével- oktatóvideók kivitelezése
2016–2017	Szerződés megkötése a Kolozsvári Magyar Operaház Jakab Melinda koreográfussal, aki megvalósította a visszalépéses keresési algoritmus koreográfiáját balett tánc segítségével
2016–2017	Szerződés megkötése a Marosvásárhelyi Művészeti Egyetem igazgatójával, András Lóránt úrral, aki megvalósította a keresési algoritmusok koreográfiáját flamenco tánc segítségével

Részvétel szakmai továbbképzéseken

Időpont	Továbbképzés adatai
2019. május	Szövegszerkesztés emelt szinten - Dr. Csernoch Mária, Debreceni Egyetem, Informatika Kar
17. szeptember	Algoritmusok és sémák az informatika oktatásában - Dr. Csernoch Mária, Debreceni Egyetem, Informatika Kar
16. november	Táblázatkezelés csupán egy tucatfüggvénnyel - Dr. Csernoch Mária, Debreceni Egyetem, Informatika Kar

XXII. Reál- és humántudományi Erdélyi Tudományos Diákköri Konferencia bemutatott dolgozat vezetése

Időpont

Dolgozat címe

2019. május

Algo-Ritmika Új dimenziók (különdíj)

Szerzők: Osztian Pálma Rozália, Nagy Eszter
Jáhel, Cosma Cristián

Témavezetők: Dr. Kátai Zoltán, Osztian Erika