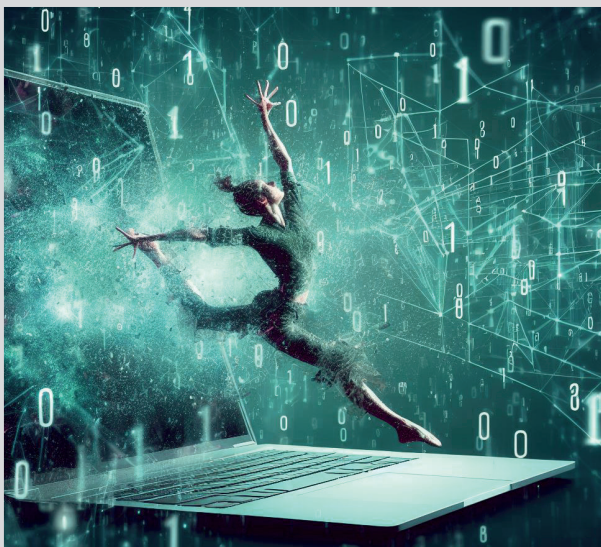




TERMÉSZETTUDOMÁNY



OSZTIÁN ERIKA

EMBERKÖZPONTÚ ALGORITMUSVIZUALIZÁCIÓ

OSZTIÁN ERIKA

*EMBERKÖZPONTÚ
ALGORITMUSVIZUALIZÁCIÓ*



SAPIENTIA KÖNYVEK



SAPIENTIA
ERDÉLYI MAGYAR
TUDOMÁNYEGYETEM

EMBERKÖZPONTÚ ALGORITMUSVIZUALIZÁCIÓ

OSZTIÁN ERIKA

| Scientia Kiadó |
| Kolozsvár · 2023 |

SAPIENTIA KÖNYVEK

Természettudományok



Kiadja a

Scientia Kiadó

40 0112 Kolozsvár, Mátyás király (Matei Corvin) u. 4.

Tel./fax: +40-364-40 1454, e-mail: scientia@kpi.sapientia.ro

www.scientiakiado.ro

Témavezető tanár: Dr. Fazekas Gábor

A doktori bizottság tagjai:

Dr. Hajdu Lajos egyetemi tanár, Debrecen

Dr. Maksa Gyula emeritus professzor, Debrecen

Dr. Muzsnay Zoltán egyetemi docens, Debrecen

Dr. Aszalós László egyetemi docens, Debrecen

Dr. Király Sándor főiskolai docens, Eger

Dr. Zsakó László egyetemi docens, Budapest

A védes ideje és helye: 2021. május 18., Debreceni Egyetem Természettudományi és Informatikai Doktori Iskola

Felelős kiadó:

Sorbán Angella

Borítóterv:

Tipotéka Kft.

Kiadói koordinátor:

Szabó Beáta

A szakmai felelősséget teljes mértékben a szerző vállalja.

Első magyar nyelvű kiadás: 2023

© Scientia 2023

Minden jog fenntartva, beleértve a sokszorosítás, a nyilvános előadás, a rádió- és televízióadás, valamint a fordítás jogát, az egyes fejezeteket illetően is.

Descrierea CIP a Bibliotecii Naționale a României

OSZTIÁN, ERIKA

Emberközpontú algoritmusvizualizáció / Osztían Erika. - Cluj-Napoca :

Scientia, 2023

Conține bibliografie

ISBN 978-606-975-085-8

007

51

004

TARTALOMJEGYZÉK

Előszó.	13
1. Bevezető.	15
1.1. Számítógépes gondolkodás a 21. században	15
1.2. Célkitűzések	16
1.2.1. Első kutatás	18
1.2.2. Második kutatás	18
1.2.3. Harmadik kutatás	18
1.2.4. Negyedik kutatás.	19
1.2.5. Ötödik kutatás.	19
1.3. A számítógépes gondolkodás fejlesztése. Kutatások	19
2. Az AlgoRhythmics kutatócsoport és projektjeik	21
2.1. A kutatócsoport bemutatása	21
Kutatásvezető.	21
Tagok	22
2.2. A kutatócsoport jelentős mérföldkövei	22
2.2.1. 2007–2009.	22
2.2.2. 2010–2015.	23
„Látni”, „hallani” és „tapintani” az algoritmusokat.	23
Algo-ritmika (eltáncolt algoritmusok).	24
Egy webes alkalmazás az algoritmusok oktatásához	26
Algoritmika mindenkinek	27
2.2.3. 2016–2021	27
Az AlgoRhythmics-projekt új lendületet kap	27
3. Az Algorythmics-környezet részletes bemutatása	29
3.1. A rendszer leírása	30
3.2. A rendszer alapvető elemei	31
3.2.1. Tanulási lépések	31
Videótanulási lépés	32
Animációtanulási lépés.	33
Levezényléstanulási lépés.	33
Kódepítés tanulási lépés	34
Életre kelt kód tanulási lépés	35
Kvíz tanulási lépés.	36
3.2.2. Algoritmusok.	37
3.2.3. Tanfolyamok	39
3.3. Múlt, jelen és jövő.	40

4. A kutatások részletes leírása	42
4.1. Sematikus versus emberi mozgást alkalmazó realisztikus algoritmusvizualizáció	42
4.1.1. Szakirodalmi áttekintés	42
4.1.2. Emberimozgás-effektus	43
4.1.3. „Egyszínű animációk” versus sokszínű táncok	44
4.1.4. Dekorációs versus reprezentációs figyelemelterelő elemek	45
4.1.5. Kutatási kérdések	46
4.1.6. Kísérlet	46
A kutatás résztvevői	46
A kutatás megvalósításának eszközei	47
4.1.7. Eredmények és kiértékelés	47
4.1.8. Absztrakt animációk versus tánckoreográfiák	48
4.1.9. Lányok versus fiúk	52
4.1.10. Következtetések	52
4.2. Sematikus és realisztikus algoritmusvizualizációk ismételt megtekintése, illetve ezek kombinálása	53
4.2.1. Szakirodalmi áttekintés	53
4.2.2. Sematikus és realisztikus dinamikus vizualizációk kombinálása	54
4.2.3. Kutatási kérdések	55
4.2.4. Kísérlet	55
A kutatás résztvevői	56
A kutatás megvalósításának eszközei	56
4.2.5. Eredmények és kiértékelés	56
4.2.6. Következtetések	59
4.3. Az interaktivitás szerepe online algoritmusvizualizációs környezetekben	60
4.3.1. Szakirodalmi áttekintés	60
4.3.2. Kutatási kérdések	62
4.3.3. Kísérlet	62
A kutatás résztvevői	62
A kutatás megvalósításának eszközei	63
4.3.4. Eredmények és kiértékelés	63
4.3.5. Következtetések	67
4.4. Tanári kérdéssorok algoritmusvizualizációs környezetekben	67
4.4.1. Szakirodalmi áttekintés	67
4.4.2. Kutatási kérdések	68
4.4.3. Kísérlet	69
A kutatás résztvevői	69
A kutatás megvalósításának eszközei	69
4.4.4. Eredmények és kiértékelés	71
4.4.5. Következtetések	74

4.5. Számítógépes gondolkodás vizsgálata különböző korosztályú tanulóknál	74
4.5.1. Szakirodalmi áttekintés	74
4.5.2. Az AlgoRythmics-ötletre épülő számítógépes gondolkodási teszt . . .	75
4.5.3. Kutatási kérdések	76
4.5.4. Kísérlet	77
A kutatás résztvevői.	77
A kutatás megvalósításának eszközei	78
4.5.5. Eredmények és kiértékelés	78
4.5.6. Korosztályonkénti elemzés.	79
4.5.7. Iskolánkénti elemzés.	82
4.5.8. Nemek szerinti elemzés	84
4.5.9. Következtetések.	85
5. A kutatások összegzése	87
5.1. Következtetések.	87
6. Következtetések.	90
7. Mellékletek	91
7.1. A kutatások menete.	91
7.1.1. Első kutatás menete	91
7.1.2. Második kutatás menete	93
7.1.3. Harmadik kutatás menete.	95
7.1.4. Negyedik kutatás menete	96
7.1.5. Ötödik kutatás menete	99
Függelék.	103
Köszönetnyilvánítás	105
Ábrajegyzék	107
Táblázatjegyzék	109
Irodalomjegyzék	111
Forrásjegyzék.	119
Rezumat	121
Abstract	123
A szerzőről.	125
Köszönetnyilvánítás	127

CUPRINS

Prefață	13
1. Introducere	15
1.1. Gândirea computațională în secolul al XXI-lea	15
1.2. Obiective	16
1.3. Dezvoltarea gândirii computaționale. Studii	19
2. Grup de cercetare AlgoRythmics. Proiecte	21
2.1. Prezentare generală	21
2.2. Activitatea grupului de cercetare	23
3. Prezentarea mediului AlgoRythmics	29
3.1. Prezentarea sistemului	30
3.2. Elementele de bază ale sistemului	31
3.3. Trecut, prezent, viitor	40
4. Descrierea detaliată a cercetării	42
4.1. Vizualizarea realistă a algoritmilor utilizând mișcarea schematică versus umană	42
4.2. Vizualizări semantice și realiste ale algoritmilor: repetarea și combinarea acestora	53
4.3. Rolul interactivității în mediul de vizualizare a algoritmilor online	60
4.4. Set de întrebări de la profesori către elevi în medii de vizualizare a algoritmilor	67
4.5. Studiul gândirii computaționale la elevi de vârste diferite	74
5. Rezumatul studiilor	87
5.1. Concluzii	87
6. Concluzii	90
7. Anexe	91
Appendix	103
Lista figurilor	107
Lista tabelelor	109
Bibliografie	111
Lista de surse	119
Rezumat	121
Despre autor	125
Mulțumiri	127

CONTENTS

Foreword	13
1. Introduction.	15
1.1. Computational thinking in the 21 st century.	15
1.2. Goals	16
1.3. Enhancing computational thinking. Studies	19
2. AlgoRythmics research group. Projects	21
2.1. Overview	21
2.2. The research group's activity	22
3. AlgoRythmics environment	29
3.1. Description of the system	30
3.2. The fundamental elements of the system	31
3.3. Past, present, future	40
4. A detailed description of research	42
4.1. Realistic algorithm visualization using schematic versus human motion	42
4.2. A revisit of semantic and realistic algorithm visualizations, as well as their combination.	53
4.3. The role of interactivity in online algorithm visualization environment	60
4.4. Teaching question sets in algorithm visualization environments.	67
4.5. Examining students' computational thinking at different ages	74
5. Summary	87
5.1. Conclusions.	87
6. Conclusions	90
7. Attachments.	91
Appendix.	103
List of figures	107
List of tables.	109
Bibliography	111
List of sources	119
Abstract	123
About the author	125
Acknowledgements.	127

ELŐSZÓ

A könyv a szerző *Interaktív, többérzékszerves elektronikus oktatási környezet a számítógépes gondolkodás fejlesztéséhez* c. doktori értekezésének szerkesztett változata, amelyet a Debreceni Egyetem Matematika- és Számítástudományok Doktori Iskola tanácsa előtt védett meg 2021 májusában.

Célja, hogy bepillantást nyújtson a Sapientia Erdélyi Magyar Tudományegyetem által kifejlesztett *AlgoRhythmic*s tanulási környezetbe, illetve az ezzel kapcsolatos kutatásokba, amelyet az informatika oktatásának, valamint a számítógépes gondolkodás fejlesztésének érdekében alkottak meg az intézmény oktatói.

Az *AlgoRhythmic*s tanulási környezet nemcsak a rendezési, hanem a keresési algoritmusokat is tánckoreográfiákkal ábrázolja, kiemelve a sematikus és realisztikus vizualizációk közötti különbségeket. Az ábrázolásokat az etnikumok néptáncai gazdagítják. A könnyen érthető stílus, az ábrák és képek, valamint a táblázatok nagymértékben elősegítik a témák megértését.

A kiadvány részletesen ismerteti az *AlgoRhythmic*s webes környezet fejlődését, beleértve az interaktivitási szint beállításának lehetőségét is. A mellékletek, valamint az irodalomjegyzék további információkat nyújtanak azok számára, akik az algoritmusvizualizációval és az ezzel kapcsolatos kutatásokkal mélyebb szinten szeretnének foglalkozni. Informatika szakos hallgatók, oktatók, számítógépes szakemberek, illetve a téma iránt érdeklődők egyaránt haszonnal forgathatják.

Osztaián Erika

Marosvásárhely, 2023. november 27.

1. BEVEZETŐ

1.1. Számítógépes gondolkodás a 21. században

A 20. század második fele már jelentős mértékben átformálta a társadalmat, ahogy a számítógépek fokozatosan teret hódítottak. Azonban a jelenlegi állapotban ez a folyamat olyan magas szintre emelkedett, hogy a számítógépes gondolkodás a társadalom minden tagja számára alapkészség kellene legyen. Wing 2006-ban írt jelentős cikkében úgy fogalmazott, hogy a számítógépes gondolkodás mint új műveltség válik a 21. század meghatározó oktatási elvévé és alapvető tanulási képességévé (Wing 2006). Ennek köszönhetően elindult egy „Számítógépes gondolkodás mindenkinek” mozgalom, amelynek egyik közvetlen eredménye az, hogy az oktatás minden szintjén (az elemi, gimnáziumi, középiskolai és felsőoktatási szinteken) erőfeszítések történtek a számítógépes gondolkodás képességének fejlesztésére.

Ennek a mozgalomnak az általános céltudatossága, hogy minden ember képes legyen hatékonyan és kreatívan használni a számítógépeket és az azok által nyújtott lehetőségeket. Az oktatási intézmények egyre inkább felismerik a digitális készségek fontosságát, és aktívan dolgoznak azon, hogy a diákok megkapják az ehhez szükséges tudást és képességeket. Az ilyen törekvések révén a társadalom minden tagja jobban felkészülhet a digitális kor kihívásaira és lehetőségeire. A számítógépes gondolkodás nem csupán egy szűk réteg privilégiuma, hanem a közös jövőnk építőköve, amelynek fejlesztésére közösen kell törekednünk.

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) újraértelmezi 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.” (A számítógépes gondolkodás azt jelenti, hogy a bonyolult problémákat kisebb, kezelhető részekre bontjuk, és a megoldásokat algoritmusok vagy lépésről lépésre követhető utasítások formájában fejezzük ki, amelyeket egy számítógép tud követni.) 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, 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 lehet. Például az AlgoRythmics kutatócsoportunk, amely a Sapientia Erdélyi Magyar Tudományegyetem keretében működik, a megoldást az informatika és a tánc párosításában látja. Művészegyüttessékek karöltve tánckoreográfiákat dolgoztunk ki, amelyek alapvető számítógépes algoritmusokat illusztrálnak.

Az AlgoRythmics tanulási környezet mint webalkalmazás jelenleg hét rendezési és három keresési stratégiához tartalmaz dinamikus vizualizációkat: algoritmustáncokat és számítógépes animációkat. Amíg a tánckoreográfiák hozzájárulnak a több érzékszerves 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 AlgoRythmics platform valóban egy interaktív, több érzékszerves elektronikus oktatási környezet a számítógépes gondolkodás fejlesztéséhez.

A fent megfogalmazott címben tömörített kutatási témát szem előtt tartva, a következő célokat tűztük ki:

1.2. Célkitűzések

Amíg az absztrakt animációk a sematikus ábrázolások kategóriájába sorolhatók, addig a tánckoreográfiákkal való ábrázolás inkább egyfajta realiztikus vizualizációnak tekinthető. Korábbi tanulmányok arra 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 továbbiakban bemutatott *első* kutatás a „Sematikus versus emberi mozgást alkalmazó realisztikus algoritmusvizualizáció” 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 AlgoRhythmic környezet viszont az egyszerű megismétlés mellett azt is lehetővé teszi, hogy a vizsgált algoritmust más-más ábrázolásban tekintsék meg a diákok, ismételten. Ezért a *második* kérdés, amit megvizsgálunk, az, hogy mi az egymáshoz viszonyított értékük különböző vizualizációkombinációknak (a sematikus követi a realisztikus; a 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 alapelvre épülnek az interaktív oktatási környezetek. Az algoritmusvizualizációs eszközök általában kétféleképpen aktiválják a tanulót. Egyik módszer az, 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 felhasználót felkérjük, 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, 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 utal, és ezeket olyan kollégáktól kaptuk, akik már évek óta használják az AlgoRhythmic 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 (Káta 2015), hogy az AlgoRhythmic tanulási környezet hatékony eszköz lehet humán érdeklődésű egyetemi hallgatók számítógépes gondolkodásának a fejlesztéséhez is. Ezután felmerült a kérdés, hogy mennyire alkalmas arra, hogy különböző korosztályú tanulókat bevezessünk a számítógépes algoritmusok világába. A jelen leírás *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 gondolkodásának fejlesztéséhez.

1.2.1. Első kutatás

Azok a realiztikus algoritmusvizualizációk (mint például az AlgoRhythmic tánc Koreográfiák), amelyek emberi mozgás révén illusztrálják az algoritmikai tartalmat, ugyanolyan hatékonyak, sőt hatékonyabbak is tudnak lenni, mint a megfelelő sematikus ábrázolások (absztrakt számítógépes animációk).

Könnyebb különbséget tenni a megtekintett algoritmusok stratégiái között, amennyiben ezeket sajátos realiztikus vizualizációk ábrázolták, mint ha sematikus ábrázolásokat alkalmaztak volna.

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

1.2.2. Második kutatás

Amennyiben ismételt megtekintést alkalmazunk, az emberi mozgáseffektust hasznosító realiztikus algoritmusvizualizációk (mint például az AlgoRhythmic videók) hatékonyabbak, mint a megfelelő sematikus ábrázolások (például absztrakt számítógépes animációk).

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

1.2.3. Harmadik kutatás

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

Nincs szignifikáns különbség a különböző interaktivitási szintekkel tanuló fiúk és lányok eredményei között.

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

1.2.4. Negyedik kutatás

A tanár által irányított kérdés–felelet megbeszélésnek döntő szerepe lehet az algoritmusvizualizációs környezetekben rejlő potenciál teljesebb kiaknázásában.

1.2.5. Ötödik kutatás

Ha egy oktatási rendszer nem fókuszál explicite a számítógépes gondolkodás fejlesztésére, akkor a többi tantárgy járulékos hozadékaként csak moderáltan fejlődik e képesség.

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

1.3. A számítógépes gondolkodás fejlesztése. Kutatások

Mivel kutatásainkat hangsúlyosan az AlgoRhythmic környezetben végeztük, ezért a könyv fejezeteiben röviden bemutatásra kerül az AlgoRhythmic projekt története. A 2. fejezet a kutatócsoportot, valamint annak eddigi munkáját mutatja be, míg a 3. fejezet pedig az AlgoRhythmic online oktatási környezetet. Ezt követően a könyv gerincét szolgáló öt kutatást fejtjük ki, majd jövőbeli terveinkre fókuszálunk. Az első négy kutatás a Sapientia EMTE Marosvásárhelyi Kara elsőéves, reál szakos 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 tanulóakra. 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 AlgoRhythmic környezet, ezért az 1–4. kísérletek elején tartottunk számukra egy rövid szemléltető bemutatót, a beszűrő rendezés tánccoreográfiáját 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 „használat” és „használat + csere” műveletek sorozata. Azt is kihangsúlyoztuk, hogy a rendezési stratégiák többfelvonásos forgatókönyveknek is tekinthetők: minden felvonásban/menetben helyére kerül

egy vagy több elem. Mivel a kísérletekre a tanévek elején került sor, azt megelőzően, hogy elkezdtek volna a Programozás I. tantárgyat, a programozói előismeretekkel nem rendelkező diákoknak ez volt az első kapcsolatuk a számítógépes algoritmusokkal.

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

2. AZ ALGORYTHMICS KUTATÓCSOPORT ÉS PROJEKTJEIK

2.1. A kutatócsoport bemutatása

Kutatások (de Koning–Tabbers 2011) bizonyították, hogy a nem hagyományos oktatási stratégiák alkalmazásával a diákok fogékonyabbá válhatnak a tanulásra. Az ilyen stratégiák közé tartozik például a tánc-koreográfiák vagy dallamok használata, amelyek a művészi elemek által hozzáadott érték révén segíthetnek a tanulásban. Ezen kívül egy gondosan kidolgozott absztrakt vizualizáció segíthet elvonatkoztatni a tanulmányozott témától lényegtelen részletektől, és a tanulók figyelmét a kulcsfontosságú algoritmusok műveleteire irányíthatja.

Az AlgoRhythmics kutatócsoport munkája éppen ezekre az elvekre épül. Az algoritmustáncok és a számítógépes animációk használatával céljuk a diákok algoritmikus gondolkodásának fejlesztése. Ezen tanulási eszközök segítségével a diákok könnyebben megérthetik és elsajátíthatják az algoritmusok lényegét, miközben élvezik a művészi kifejezés és a vizuális élmények nyújtotta szórakozást. A kutatócsoport tagjai olyan szakértők, akik rendelkeznek a tervezett kutatás témájához szükséges szakképzettséggel, korábban elvégzett munkájuk is kapcsolódik a kutatáshoz, és a közös kutatómunka során szerzett együttműködési tapasztalatuk is rendkívül hasznosnak ígérkezik. A kutatásvezető, Kátai Zoltán, nemcsak a tervezett kutatás irányításáért felel, hanem az AlgoRhythmics kutatócsoport vezetőjeként és kezdeményezőjeként is tevékenykedik.

Osztían Erika már 2016-ban csatlakozott az AlgoRhythmics kutatócsoporthoz, így hosszú ideje részt vesz a csoport munkájában. Osztían Pálma Rozália pedig 2018-ban csatlakozott a csoporthoz, így már ő is egy jó ideje aktív tagja a csapatnak.

Sántha Ágnes (szociológia és teatrológia szakos) kolléganő meghívott kutatóként csatlakozik a csapathoz, mivel szakmai tapasztalatai jelentősen kiegészítik a csoport többi tagjának tudományos munkásságát, különösen a kutatás művészeti és társadalomtudományi vonatkozásaiban. Az ő szakértelme értékes hozzájárulást jelent a kutatás szempontjából.

Kutatásvezető

Dr. Kátai Zoltán gazdag munkássággal rendelkezik a tervezett kutatás témakörében. Már 2008-ban publikált arról, hogy miként lehet innovatív módon bevonni az érzékszerveket (látás, hallás, tapintás) elemi algoritmusok oktatásába (Kátai–Juhász–Adorjáni 2008). Egy következő lépés a kinesztetikus érzék bevonása volt, szerepalkítás révén, a rekurzív algoritmusok tanításába (Kátai 2011). Ezután következett

a rendezési algoritmusokat illusztráló tánckoreográfiák megalkotása, először amatőr táncosok (Kátai–Tóth 2010), majd professzionális együttesek bevonásával (Kátai 2014a; 2014b). A gyűjtemény 2018-ban bővült a keresési algoritmusokkal. A megjelent rangos publikációk mindenike olyan kutatásokról számolt be, amelyek igazolták a kidolgozott eszközök és módszerek hatékonyságát az informatikaoktatásban. Az első lépés az „algoritmika mindenkinek” irányba 2015-ben történt. A kapcsolódó kutatás humán és reál szakos hallgatók teljesítményét vizsgálta AlgoRythmics környezetben (Kátai 2015). Egy friss publikáció hasonló összehasonlító elemzésről számol be, motivációs szempontok alapján (Kátai 2020). A kutatásvezető egyéb kapcsolódó munkássága összefonódik a tagok kutató tevékenységével (lásd lentebb).

Tagok

Az Osztián Erika PhD-dolgozatában bemutatott kutatások az AlgoRythmics környezet bevonásával valósultak meg. Például az „algoritmika mindenkinek” szellemében vizsgálta az AlgoRythmics-megközelítés hatékonyságát különböző korosztályú diákok számítógépes gondolkodásának fejlesztése szempontjából, reál és művészeti iskolákban (Kátai–Osztián–Lőrincz 2021). Egy másik téma kör, amelyet körbejárt, hogy mi a tanár szerepe az AlgoRythmics-környezetben. Például hogyan segíthetnek a tanár gondosan megfogalmazott kérdései abban, hogy a tanulók érzékeljenek olyan, a számítógépes gondolkodás szempontjából fontos elemeket is, amelyek talán elkerülnék a figyelmüket, amennyiben csak magukban elemzik az algoritmusábrázolásokat (Kátai–Osztián 2021).

Osztián Pálma Rozália doktorandusz először mint fejlesztő kapcsolódott be a projektbe, és vállalt oroszlánrészt az AlgoRythmics webalkalmazás (algorythmics.com) létrehozásában. Mint fiatal kutató az interaktivitás fontosságának vizsgálata irányába mozdult el. Egyik legjelentősebb munkájában azt elemzi társszerzőivel együtt, hogy létezik-e optimális interaktivitási szint algoritmusvizualizációs környezetekben (Osztián–Kátai–Osztián 2020). Doktori dolgozatának témája is jelentősen összecseng a jelen kutatással, hisz azt vizsgálja például, hogy miként lehetne növelni az interaktivitás pozitív hatását a tanulásra a „social belonging” effektus, valamint a játékosítás alkalmazása által.

2.2. A kutatócsoport jelentős mérföldkövei

2.2.1. 2007–2009

Az AlgoRythmics 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” – vallotta Comenius, a modern oktatás atyja (Pukánszky 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 több érzékszerves tanulás elvére épült. Az építkezés lépésről lépésre, érzékszervről érzékszervre, modulárisan történt. Mivel az aktív tanulás elvét is figyelembe vettük, a kezdeti ötlet egy több érzékszerves interaktív rendszerré fejlődött.

2.2.2. 2010–2015

Dr. Kátai Zoltán gazdag munkásságának köszönhetően egy következő lépés a kinesztetikus érzék bevonása volt, szerepalakítás révén, a rekurzív algoritmusok tanításába (Kátai 2011).

„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 AlgoRhythmic kutatócsoport azonban ennél többet szeretett volna: bevonni a hallást és a tapintást is a programozásoktatásba. 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 „megszóltatásához” a kutatócsoport kifejlesztettek egy szoftvereszközt (Kátai 2014), amely hangjegysort társított az elemi algoritmusok kulcslépéseihez. 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).

Algo-ritmika (eltáncolt algoritmusok)

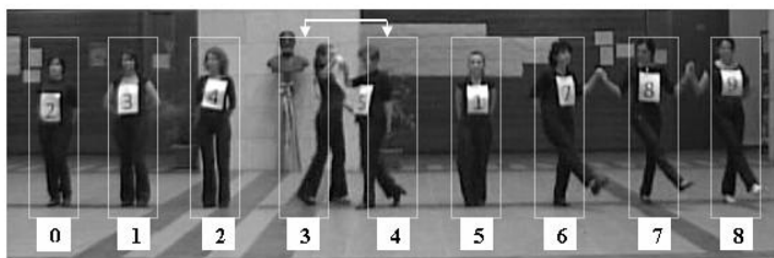
Egy következő lépés az volt, hogy miként lehetne a kinesztetikus érzékelést is bevonni a tanulás folyamatába. Mivelhogy az algoritmus egy jól meghatározott műveletsort jelent, miért is ne lehetne felfogni úgy, mint egy forgatókönyvet, majd színre vinni. Ennek az ötletnek az implementálásaként a kutatócsoport 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 (lásd 1. ábra).



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).



2. ábra. Buborékrendezés

**Quick-sort with Hungarian (Küküllőmenti legényes) folk dance**

AlgoRhythms • 1.9M views • 8 years ago

Created at Sapientia University, Tîrgu Mures (Marosvásárhely), Romania. Directed by Kátai Zoltán and Tóth László. In ...

**Merge-sort with Transylvanian-saxon (German) folk dance**

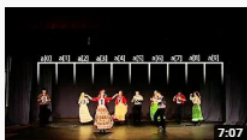
AlgoRhythms • 593K views • 8 years ago

Created at Sapientia University, Tîrgu Mures (Marosvásárhely), Romania. Directed by Kátai Zoltán and Tóth László. In ...

**Shell-sort with Hungarian (Székely) folk dance**

AlgoRhythms • 508K views • 8 years ago

Created at Sapientia University, Tîrgu Mures (Marosvásárhely), Romania. Directed by Kátai Zoltán and Tóth László. In ...

**Select-sort with Gypsy folk dance**

AlgoRhythms • 616K views • 8 years ago

Created at Sapientia University, Tîrgu Mures (Marosvásárhely), Romania. Directed by Kátai Zoltán and Tóth László. In ...

**Bubble-sort with Hungarian (\"Csángó\") folk dance**

AlgoRhythms • 1.7M views • 8 years ago

Created at Sapientia University, Tîrgu Mures (Marosvásárhely), Romania. Directed by Kátai Zoltán and Tóth László. In ...

**Insert-sort with Romanian folk dance**

AlgoRhythms • 706K views • 8 years ago

Created at Sapientia University, Tîrgu Mures (Marosvásárhely), Romania. Directed by Kátai Zoltán and Tóth László. In ...

3. ábra. Rendezési algoritmusok videói

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

Innen már csak egy lépés volt a ma már jól ismert AlgoRhythmics táncokig. A kutatócsoport együttműködést kezdeményezett a Maros művészegyettesel (néptáncosokkal), akik segítségével a rendezési algoritmusokat illusztráló tánc-koreográfiákat művészileg is színvonalassá tették. Ezen együttműködés gyümölcseként született meg 2011-ben hat rendezési algoritmus videóra vett tánc-koreográfia illusztrációja (3. ábra).

A YouTube-on (www.youtube.com/user/AlgoRhythmics) 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.

Egy webes alkalmazás az algoritmusok oktatásához

A kutatók kezdetektől tisztában voltak azzal, hogy bár a tánc-koreográfia-ábrázolásoknak meglehetnek 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 AlgoRhythmics webes környezetet (<https://www.youtube.com/user/AlgoRhythmics>), ahol minden videó mellé egy absztrakt animációt társítottak. Ezen új modul további erőssége az volt, hogy interaktív üzemmódban 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 AlgoRhythmics-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” (Kátai 2014). E tanulságot úgy kamatoztattuk a későbbi fejlesztéseknél, hogy néhány újabb tánc-koreográfiában a szereplők csak a hátukon viselik a számokat (szemből nézve őket, a számsorozat rejtett).

Algoritmika mindenkinek

Végül az AlgoRhythmics alkalmazás több lett, mint több érzékszerves interaktív tanulási környezet. Talán legszembeütőbb 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 (Kátai–Adorjáni 2014; Kátai 2015).

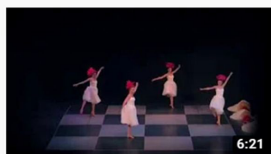
Ami az alkalmazott táncstípusokat illeti, a kutatók az Erdélyben honos etnikumok néptáncaiból merítettek: magyar, székely, román, cigány, szász, csángó. Mivel mindezekhez természetesen nemzeti érzület is kapcsolódik, egy olyan kutatás is született, amely azt vizsgálta, hogy befolyásolja-e mindez a tanulás folyamatát. Érdekes eredmények születtek, amikor román, magyar és cigány néptáncok által illusztrált algoritmusokat alkalmaztak tiszta magyar, tiszta román és vegyes oktatási környezetben (Kátai 2014b).

2.2.3. 2016–2021

Az AlgoRhythmics-projekt új lendületet kap

2016-tól az AlgoRhythmics-projekt új lendületet kapott. Ekkor csatlakozott a kutatócsoporthoz Osztian Erika. Döntés született, hogy a projektet az alábbi négy dimenzió mentén fejlesszük tovább: 1. algoritmustípus; 2. táncstípus; 3. ábrázolási mód; 4. a táncról a kódig.

1. Kiterjesztettük az algoritmuskollekciót a keresési stratégiákra is: lineáris keresés, bináris keresés, visszalépéses módszer (backtracking).



BACKTRACKING ballet choreography (The Four Queens)

AlgoRhythmics • 27K views • 1 year ago

Created at Sapientia University, Targu Mures (Marosvásárhely), Romania. Directed by Kátai Zoltán, Osztian Erika, Osztian ...

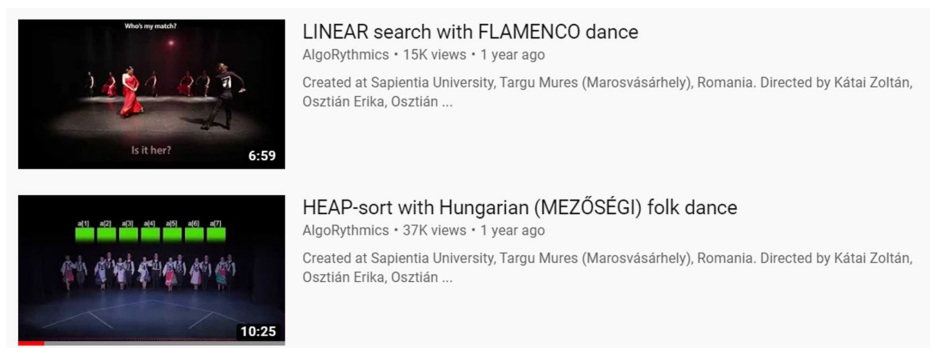


BINARY search with FLAMENCO dance

AlgoRhythmics • 56K views • 1 year ago

Created at Sapientia University, Targu Mures (Marosvásárhely), Romania. Directed by Kátai Zoltán, Osztian Erika, Osztian ...

4a. ábra. Új algoritmusok 2016-tól

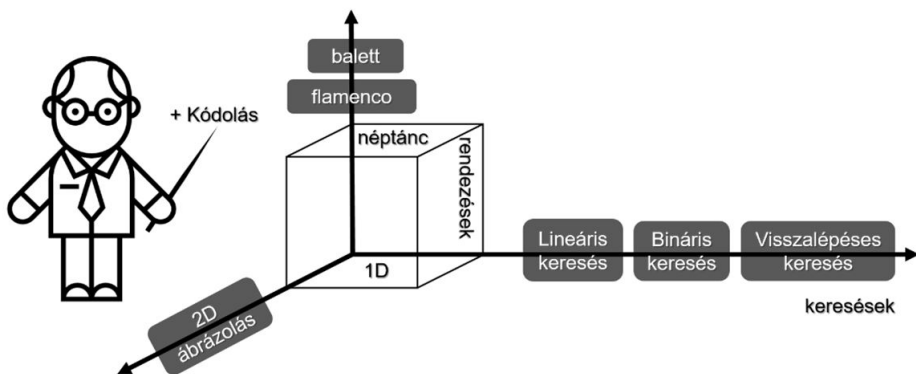


4b. ábra. Új algoritmusok 2016-tól

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 felé: flamenco és balett.

4. Teljesen újraterveztük a webes alkalmazást, amely olyan modulokkal bővült, amelyek elvezetik a diákot az algoritmusok megértésétől ezek implementálásáig (5. ábra).



5. ábra. Az AlgoRhythmics új dimenziói

3. AZ ALGORYTHMICS-KÖRNYEZET RÉSZLETES BEMUTATÁSA

Amikor az oktatás, az informatika és az algoritmusok témaköréről beszélünk, kiemelkedő jelentősége van az online oktatási környezeteknek. Napjaink oktatási stratégiái széles körben támogatják ezeknek a rendszereknek a használatát. Gyakran egy-egy hagyományos, osztályteremben bemutatott algoritmus magyarázatához társítanak kifejező animációkat vagy videókat, amelyek még inkább segítik a tanulókat az algoritmusok megértésében és elsajátításában.

Mindezt figyelembe véve úgy gondoltuk, igen hasznosnak bizonyulna egy olyan oktatási környezet biztosítása, amely segíti az alapvető algoritmusok elsajátítását. Így született meg az AlgoRythmics online oktatási környezet, melynek alapját a 2011-ben közzétett tánc-koreográfia-videók képezték. A név kettős jelentéssel bír: algoritmika és ritmus, vagyis a tudomány és művészet ötvözése, mely a mi esetünkben az informatikai stratégiákban és a táncban valósul meg. Az említett egyedi tánc-koreográfia-videók erdélyi magyar etnikumok által élvezetesen mutatják be a különböző rendező és kereső algoritmusok működését. Az idő múlásával egyértelművé vált, hogy ezek a tánc-koreográfia-videók rendkívül népszerűek, így elkezdtük továbbfejleszteni a környezetet. Ekkor jött létre az a módosított oktatási rendszer, amely a tánc-koreográfia-videók mellett animációkat is felhasznált az algoritmusok vizualizálásához.

Az elmúlt időszak során további három új tanulási lépést vezettünk be, melyeket a meglévő vizualizációkkal társítottunk. Emellett a tánc-koreográfia-videók száma is 10-re növekedett. Így kialakult egy online oktatási környezet, amely lehetőséget nyújtott az algoritmusok alapos megértésére és különféle kutatási témák kipróbálására is. Az elmúlt évek során számos kutatást végeztünk, amelyek révén új tapasztalatokra tettünk szert mind az oktatási stratégia, mind a környezet fejlesztése terén.

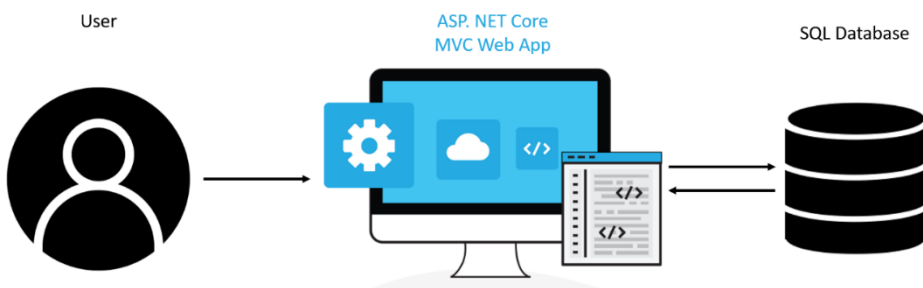
Az általunk kitűzött cél az volt, hogy 2021-ben egy teljesen új változatot hozunk létre, amely kiterjeszti a környezet elérhetőségét más intézmények oktatói számára, és lehetővé teszi személyre szabott tanfolyamok létrehozását. A nagyközönség számára új lehetőségek kapuját nyitottuk meg, melyek, úgy gondoljuk, hatással vannak az oktatásra és az innovatív tanulási megközelítések kialakítására.

A korábbi kutatási eredmények és az eddigi tapasztalatok mélyebb vizsgálata után nagy reményekkel tekintettünk az új környezet kialakítása felé. Célunk az volt, hogy segítsük a tanulókat az alapvető informatikai algoritmusok megértésében és a motiváció fenntartásában. Emellett abban is bízunk, hogy az új környezet lehetőséget biztosít más egyetemekkel és iskolákkal való kapcsolatok kiépítésére, ezzel biztosítva az oktatók és diákok közötti új és inspiráló együttműködéseket.

Ebben a fejezetben az AlgoRythmics online oktatási környezet főbb elemeit és ezek működését fogjuk bemutatni, amelyet már számos konferencián sikerült népszerűsíteni (Osztian et al. 2021).

3.1. A rendszer leírása

A rendszer architektúrájából (6. ábra) is kitűnik, hogy a felhasználók az interneten keresztül érik el a webalkalmazást. Az alkalmazás mögött egy SQL adatbázis áll, amelyen keresztül szolgáltatja az adatokat a klienseknek.

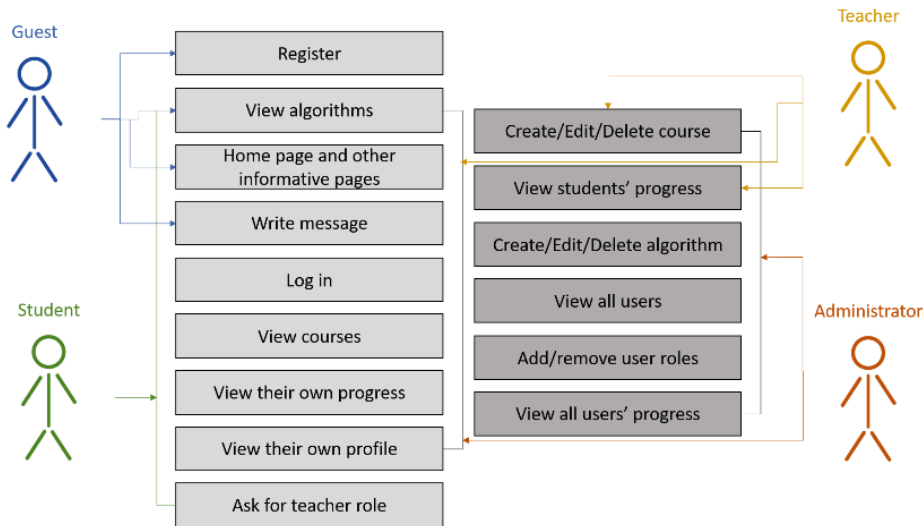


6. ábra. A rendszer architektúrájának diagramja

Az alkalmazás négy különböző felhasználói szerepkörrel rendelkezik, mint például az adminisztrátor, tanár, diák és vendég. Az alkalmazás használati eset diagramján láthatjuk, hogy a vendégfelhasználó csak az informatív oldalakat (pl. Főoldal, Elérhetőségek, Közösségi oldalak), az algoritmusok listáját és az alaptanfolyamokat érheti el, valamint bejelentkezhet (7. ábra). A bejelentkezett felhasználó automatikusan tanulói jogot kap, amely lehetővé teszi számára, hogy elvégezze az elérhető tanfolyamokat, nyomon kövesse a tanulmányi előrehaladását, és megtekinthesse saját, személyes adatait. Emellett a bejelentkezett felhasználók természetesen rendelkeznek a vendégfelhasználó által kapott jogokkal is.

A következő felhasználói szerepkör a tanároké. Bármely bejelentkezett felhasználónak, aki tanárként kívánja használni a webalkalmazást, először jeleznie kell szándékát az adminisztrátornak. Ezt megteheti az alkalmazás felületén vagy az elérhetőségek menüpontban található információk segítségével. A tanár minden előzőleg említett joggal rendelkezik, továbbá lehetősége van saját tanfolyamok létrehozására. Minden tanfolyam létrehozása során választhat a saját, rejtett és publikus tanfolyamok között. A tanár továbbá nyomon követheti a saját tanfolyamához tartozó tanulók előrehaladását.

Az adminisztrátor minden korábban említett jogot birtokol, és emellett kezelheti az algoritmusok listáját is. Ő az, aki a felhasználói jogok módosítására is jogosult, azaz aki adhat vagy elvonhat tanári jogot a felhasználóktól. További feladatai közé tartozik minden nyilvános típusú tanfolyam létrehozásának jóváhagyása.



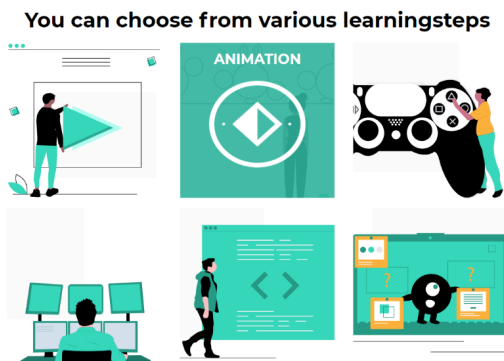
7. ábra. Használati eset diagramja

3.2. A rendszer alapvető elemei

Azáltal, hogy a webalkalmazás kreatívan és változatosan mutatja be az algoritmusokat és az azokhoz kapcsolódó tanfolyamokat, kiemelten fontos volt az alkalmazás alapvető egységeinek, mint például a tanulási lépéseknek, az algoritmusoknak és a tanfolyamoknak a pontos meghatározása és gondos kidolgozása.

3.2.1. Tanulási lépések

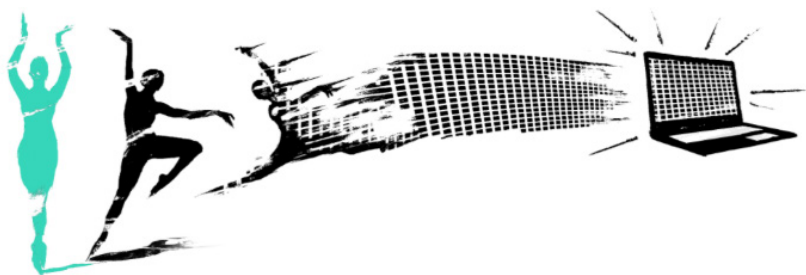
Az oldalt alkotó legalapvetőbb elemek az úgynevezett tanulási lépések. Az alkalmazás jelenleg hat alapvető tanulási lépést (8. ábra) kínál az egyes algoritmusok elsajátításához. Ezek a tanulási lépések egy ajánlott sorrend alapján vannak rendezve, melynek segítségével a felhasználók könnyebben követhetik a nehézségi szintek szerinti fejlődést. Az elrendezés két megértést, két kódolást és egy összefoglaló jellegű lépést tartalmaz annak érdekében, hogy a lehető leg-hatékonyabb tanulási tapasztalatot biztosítsa.



8. ábra. Tanulási lépések

Videótanulási lépés

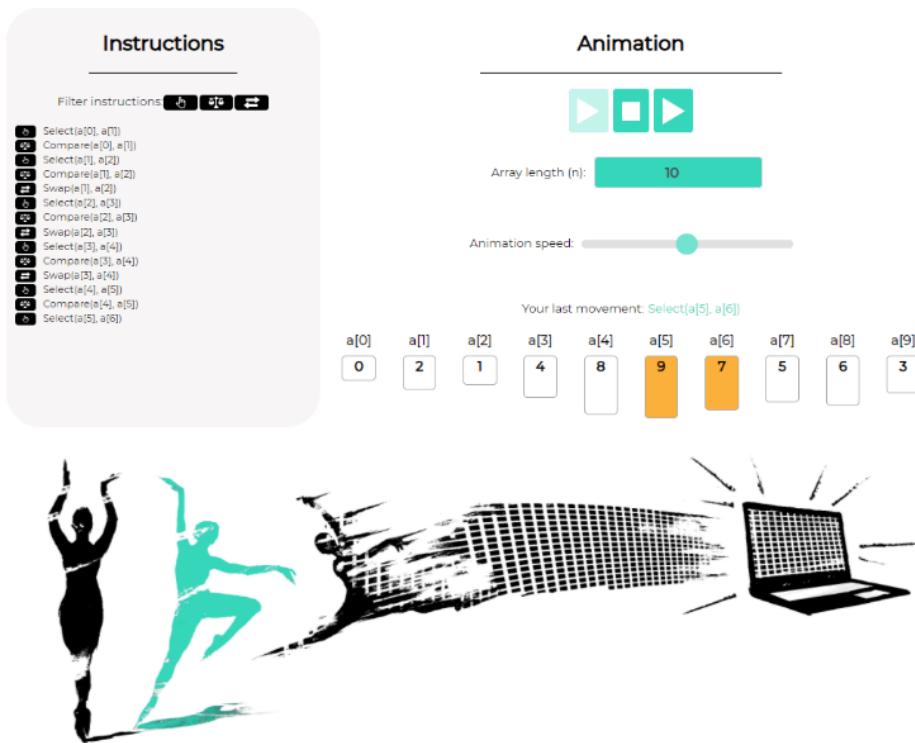
Az első tanulási lépés a videó, amely bemutatja az adott algoritmust egy tánckoreográfia segítségével (9. ábra). Alapvetően ez egy automatikusan lejátszódó videó, de a speciális, személyre szabott tanfolyamok esetén interaktív videót is hozzárendelhetünk a tanmenethez. Az interaktív videó esetén a videó előre meghatározott kulcsmomentumoknál megáll, és a felhasználónak aktívan részt kell vennie a tanulási folyamatban, hiszen válaszolnia kell a feltett kérdésekre.



9. ábra. Videótanulási lépés

Animációtanulási lépés

Az animáció az algoritmusvizualizáció egy absztraktabb formája (10. ábra), amely szintén hozzájárul az algoritmusok megértéséhez. Az animáció, hasonlóan a videóhoz, lehet automatikus vagy interaktív, és emellett meghatározható a bemeneti számsorozat jellege is (legjobb, legrosszabb eset, véletlenszerű vagy konkrét számsorozat). Továbbá, animáció létrehozásakor a tanár vagy adminisztrátor választhatja a látható (a számsorozat elemeinek értéke látható a felhasználó számára), illetve a rejtett (a számsorozat értékei rejtve maradnak a felhasználó előtt) megjelenítési módot is.

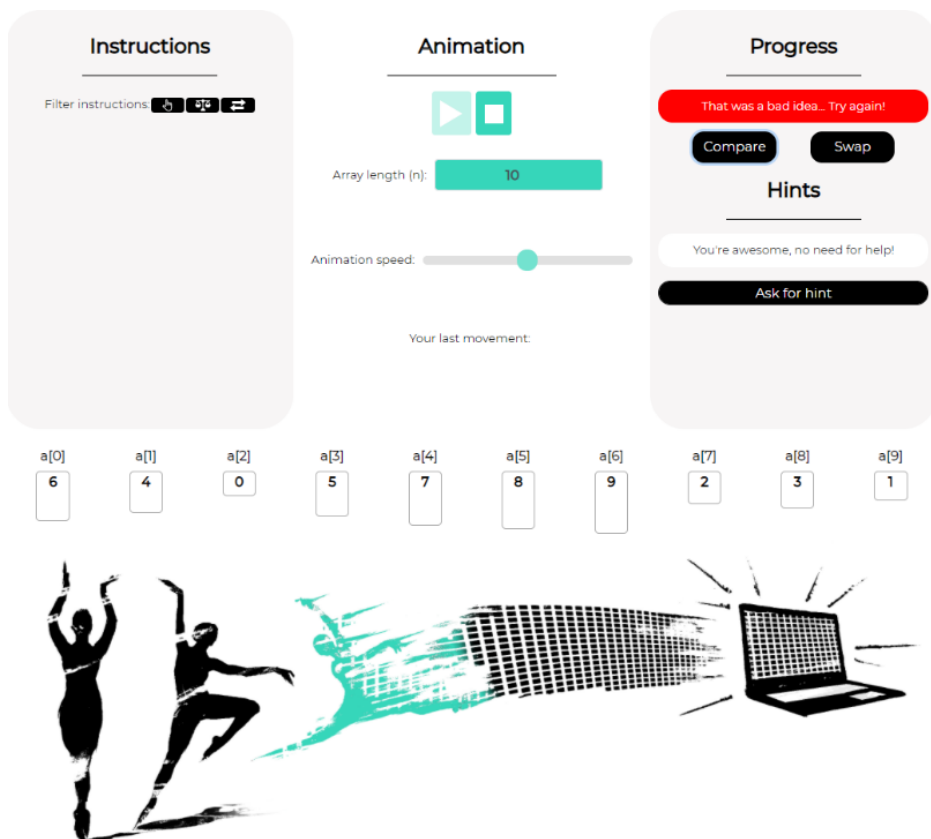


10. ábra. Animációtanulási lépés

Levezényléstanulási lépés

A levezénylés az animáció továbbfejlesztett változata, amely lehetővé teszi a felhasználó számára az algoritmushoz tartozó animáció irányítását az algoritmusstratégia elejétől a végéig (11. ábra). Ebben a módban a felhasználó maga

választhatja ki a következő összehasonlítandó vagy cserélendő elemeket, saját belátása szerint. Az algoritmus bemenetének típusától és a megjelenítési formától függően, ez a tanulási lépés is működhet különböző módokban, hasonlóan az előző tanulási lépésekhez.



11. ábra. Levezényléstanulási lépés

Kódépítés tanulási lépés

A kódépítés az első lépés a tanulási folyamatban, amikor a felhasználó számára láthatóvá válik az algoritmushoz tartozó kód C/C++ változata (12. ábra). Az animáció révén a felhasználónak lehetősége van a megjelenített kódrészletet kiegészíteni, például beállítani a ciklushatárokat vagy kiegészíteni a hiányos feltételeket. Ez a tanulási lépés már a konkrét kódolásra koncentrál, és segíti a felhasználót a kódírás stratégiája fejlesztésében.

Animation



Array length (n): 10

Animation speed:

Your last movement: Select(a[1], a[2])

Code

```

int i =   ;
do {
    last_swap_index = 0;
    for(int j =   ; j <=   ; ++j) {
    }
    i = last_swap_index;
} while (last_swap_index >   );

```

Hints

You're awesome! No need for help!

Ask for hint



a[0]

3



a[1]

7

a[2]

6

a[3]

2

a[4]

0

a[5]

4

a[6]

9

a[7]

1

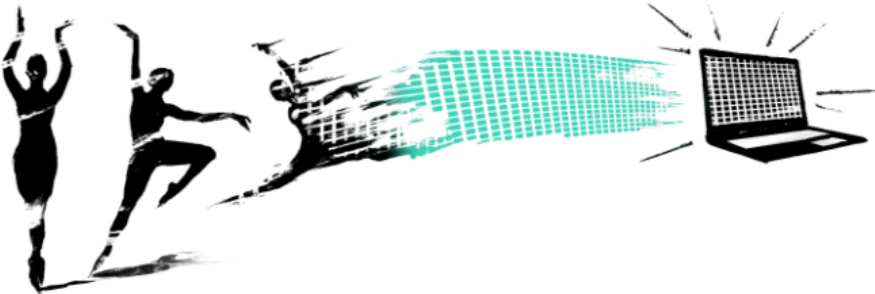
a[8]

8

a[9]

5



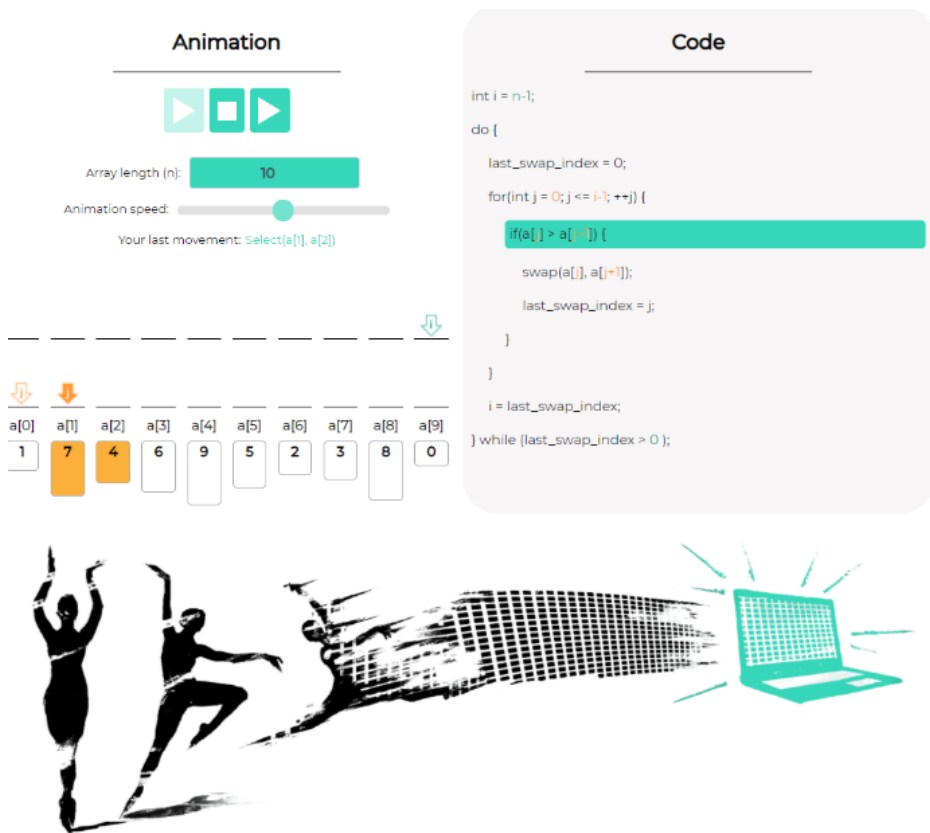


12. ábra. Kódepítés-tanulási lépés

Életre kelt kód tanulási lépés

Az életre kelt kód az előző négy tanulási lépés egyfajta összefoglalása (13. ábra). Ebben a tanulási lépésben az animáció segítségével az előző lépések során kiegészített kód „fut le”. Az animáció során a kód különböző részei hangsúlyozódnak ki az adott lépésekkel összhangban. Ez a lépés lehetővé teszi a felhasználó számára, hogy látványos módon nyomon kövesse az eddig tanult

lépések sorozatát. Az animáció és kód együttes bemutatása erősíti az algoritmus működésének megértését.



13. ábra. Életre kelt kód tanulási lépés

Kvíz tanulási lépés

A tudás felmérésére szolgáló lépés egy olyan tanulási szakasz, amelynek célja, hogy értékelje a korábban megtanult ismereteket és készségeket (14. ábra). Ez a lépés lehetővé teszi, hogy az eddigi tanulási folyamat során felhalmozódott tudást objektíven mérjük fel, és ellenőrizzük, hogy a felhasználónak milyen mértékben sikerült megérteni a bemutatott algoritmusokat és azok működését.

B course



A form of sorting by exchanging that simply interchanges pairs of elements that are out of order in a sequence of passes through the file, until no such pairs exist. The method is not competitive with straight insertion.

Clicking on this step you can see the Video learningstep of the algorithm.

Question list

1. Previous question
2. Current question
3. Third question
4. Fourth question
5. Fifth question
6. Sixth question
7. Seventh question
8. Eith question
9. Ninth question

Question

- ☒ Correct answer
- ☐ Answer 1
- ☐ Answer 2
- ☐ Answer 3
- ☐ Answer 4

Next



Submit

14. ábra. Kvíz tanulási lépés

3.2.2. Algoritmusok

A különböző algoritmusok vizualizációját az összehangolt tanulási lépések együttese biztosítja. Az alkalmazásban különféle rendező (buborék, minimum kereső, shell, beszűrő, összefésülő, gyors rendezés) és kereső algoritmusok (lineáris, bináris keresés) érhetők el (15. ábra). Emellett, az algoritmusok típusától függően, a környezet különbséget tesz iteratív és rekurzív algoritmusok megjelenítése között is. Ez az átfogó megközelítés lehetővé teszi a felhasználók számára, hogy egy megfelelő képet és stratégiai bemutatást kapjanak az algoritmusok működéséről abban a reményben, hogy ez elősegíti a programozási készségeik hatékony fejlesztését.

Algorithms

[Add new algorithm](#)

ICON	NAME	TYPE	STATUS	ACTIONS
	Bubble sort	Sorting	Unpublish	Edit Delete
	Insertion sort	Sorting	Unpublish	Edit Delete
	Selection sort	Sorting	Unpublish	Edit Delete
	Shell sort	Sorting	Unpublish	Edit Delete
	Linear search	Searching	Unpublish	Edit Delete

15. ábra. Algoritmusok listája – adminisztrátori felület

Az algoritmusok listája a rendező, kereső és visszalépéses kereső algoritmusok csoportjai szerint van felosztva (16. ábra). Amikor az egyes algoritmusok vizualizációját megtekintjük, a felhasználónak lehetősége van az algoritmushoz tartozó alaptanfolyamot is megnézni. Ez a tanulási lépések egy jól meghatározott alapsorrendjét tartalmazza, melyek megoldási sorrendje nem kötelező, azonban ajánlott, hogy az előrehaladás hatékony legyen. Mivel az alaptanfolyam vendégfelhasználó esetén is elérhető, az alaptanfolyamhoz tartozó interaktív lépések során elkövetett hibák úgynevezett ismeretlen felhasználóként kerülnek rögzítésre az adatbázisban.

All algorithms

Sorting algorithms

Searching algorithms

Backtracking algorithms



Bubble sort

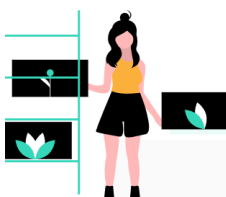
Bubble sort, sometimes referred to as sinking sort, is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements and swaps them if they are in the wrong order. The ...

[Read more](#)

Insertion sort

Insertion sort is a simple sorting algorithm that works similar to the way you sort playing cards in your hands. The array is virtually split into a sorted and an unsorted part. Values from the unsorted...

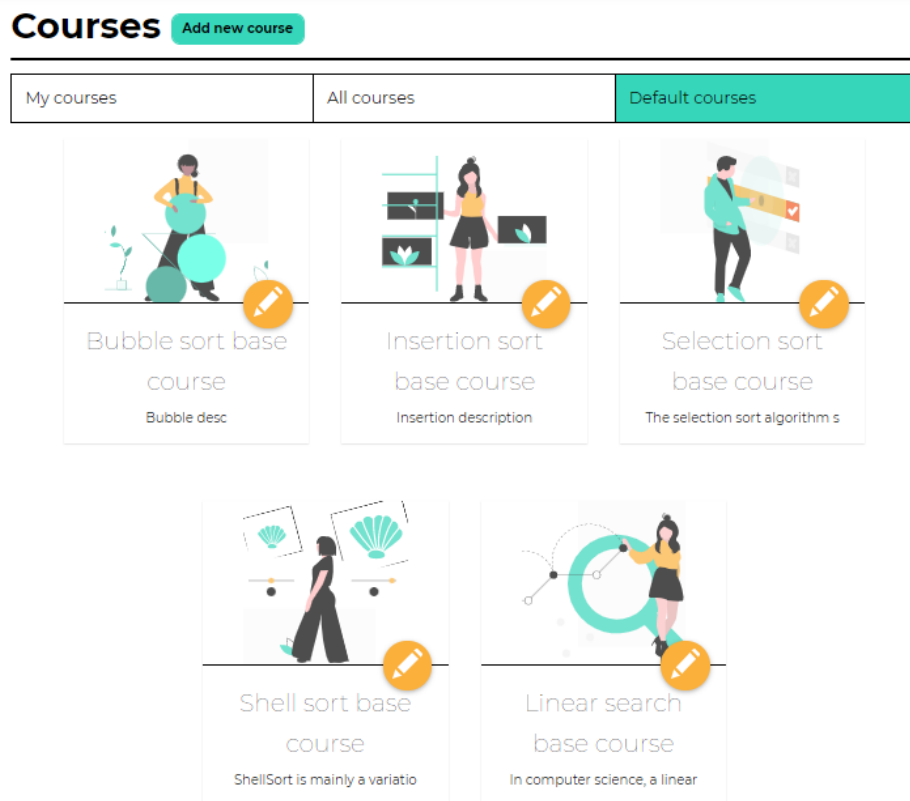
[Read more](#)



16. ábra. Algoritmusok listája felhasználói felület

3.2.3. Tanfolyamok

Ha saját, egyedi tanfolyamokat kívánunk létrehozni, segítségül szolgál a tanfolyamkészítő felület, mely elérhető mind a tanárok, mind az adminisztrátorok számára (17. ábra). A felhasználók ezen a felületen könnyedén létrehozhatnak és testre szabhatnak tanfolyamokat a saját igényeikhez és az adott tananyaghoz igazodva.



17. ábra. Tanfolyamok listája – adminisztrátori felület

Az adott tanfolyam hasonló az algoritmusok alapvető tananyagához, azonban ebben az esetben a tanár szabadon dönthet a tanulási lépések számáról, jellegéről és azok sorrendjéről. Ennek eredményeként lehetőség nyílik arra is, hogy egy tanfolyam során kizárólag különböző animációkat vagy éppen a kódépítést alkalmazza az oktató a tanulási folyamat segítésére. A rugalmasság lehetővé teszi, hogy az oktatási anyagokat az adott csoport vagy hallgatók igényeihez igazítsa a tanár (18. ábra).

Course id:
Course password:

Popular courses

All courses

Sorting courses

Searching courses

Backtracking courses

All courses

Linear search course

Linear search course description

Bubble sort course

Bubble sort course description

Selection sort course

Selection sort course description

Insertion sort course

Insertion sort course description

Shell sort course

Shell sort course description

18. ábra. Tanfolyamok listája – felhasználói felület

3.3. Múlt, jelen és jövő

Az alkalmazás jelen változata interaktív online oktatási környezetet kínál, amely lehetőséget ad az informatikai algoritmusok tanulására. A felhasználók különböző tanulási lépéseken keresztül sajátíthatják el a rendező és kereső stratégiák működését és azok alkalmazását. A tanulási lépések magukban foglalják a tánccoreográfia-videókat, -animációkat, a leveleznylést, valamint a kódépítés és életre kelt kód fázisát, melyek segítik a megértést és a kódolási képességek

fejlesztését is. Az objektív felmérés érdekében az alkalmazás kvízjellegű tanulási lépés elvégzését is biztosítja.

Továbbá, az algoritmusok mellett, hogy stratégiai szempontból külön csoportokba vannak osztva (rendező és kereső), a működési elv szerint is két osztályba sorolhatók iteratív és rekurzív jelleg szerint. A felhasználóknak lehetőségük van saját tanfolyamok létrehozására, ahol szabadon alakíthatják a tanulási lépések számát és sorrendjét, így olyan tananyagot biztosítva, amely a hallgatók igényeihez igazodik.

Az alkalmazás nemcsak az algoritmusok hatékony elsajátítására összpontosít, hanem lehetőséget biztosít az oktatóknak is a tanfolyamok egyedi megalkotására és személyre szabott kialakítására. Az ismeretlen felhasználóként rögzített hibák segítenek a folyamatos fejlődésben és az oktatási anyagok javításában is.

Összességében az alkalmazás sokoldalú és interaktív tanulási élményt nyújt, amely elősegíti az informatikai tudás megszerzését és gyakorlását, segítve a felhasználókat a programozási készségek fejlesztésében és az algoritmusok sikeres alkalmazásában. Eddigi munkánk során a környezet lehetőséget biztosított számos kutatás megvalósítására, melynek középpontjában a különféle algoritmus-vizualizációk álltak. A jelen oktatási környezet legfrissebb változata a következő publikus weboldalon érhető el: www.algo-rhythmic.com.

4. A KUTATÁSOK RÉSZLETES LEÍRÁSA

A kutatások tartalma öt nagy gondolatra épült, amelyek az eddigi 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	<i>Acta Polytechnica Hungarica, Journal of Applied Sciences</i>
Schematic versus human movement effect enriched realistic dynamic visualization	<i>Journal of Computer Assisted Learning</i>
Algorithm visualization environments: can an optimal interactivity level be established?	<i>Frontiers in Education</i>
Improving AlgoRythmics Teaching Learning Environment by Asking Questions	<i>International Journal of Instruction</i>
Investigating young school students' computational thinking ability across grade levels	<i>Journal of Educational Computing Research</i>

A következő öt alfejezetben ezek a kutatások kerülnek bemutatásra.

4.1. Sematikus versus emberi mozgást alkalmazó realisztikus algoritmusvizualizáció

4.1.1. Szakirodalmi áttekintés

Amint már említettük, a számítógépes algoritmusoknak kiemelt szerepük van a tanulók számítógépes gondolkodásának fejlesztésében. Mivel az algoritmusok elvont folyamatok, a vizualizáció 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 (számítógép nélküli) informatikaoktatás keretében, amelyek közelebb állnak a realisztikus ábrázoláshoz. Egy ilyen példa az AlgoRhythms (2020) környezet is, amely realisztikus tánckoreográfiákkal illusztrál számítógépes algoritmusokat.

Az eddigi kutatási eredmények azt mutatják, hogy mind a kétfajta ábrázolási módnak meglehetnek az előnyei attól függően, hogy mi a szem előtt tartott tanulási cél. Például a sematikus ábrázolások egyik fontos előnye, hogy segítenek a tanulónak 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.

4.1.2. Emberimozgás-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, mint ha 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 „emberimozgás-effektusnak” („human movement effect”) nevezi (Shaffer et al. 2010).

Bár mind a realisztikus, mint a sematikus ábrázolásoknak megvannak 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 kétszer tekintette meg ugyanazt a realisztikus ábrázolást, a második csoport kétszer 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 tekintette meg, a második csak a realisztikust, a harmadik csoport először a sematikust, majd a realisztikust, egymást követően, a negyedik pedig párhuzamosan követhette a képernyő bal oldalán a sematikus, a jobb oldalán 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 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 AlgoRythmics környezetre, ahol a realisztikus ábrázolásokat a tánc Koreográfiák, a sematikusakat pedig a számítógépes animációk képviselik.

4.1.3. „Egyszínű animációk” versus sokszínű táncok

Az AlgoRythmics 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, színt váltanak.

Ezzel szemben mindenik tánc Koreográfiának egyedi stílusa van, amely a felhasznált táncstípusok 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 úgy mond 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 tánckoreográfiákkal tanulták az algoritmusokat, a visszaemlékezéskor könnyebben különbséget tudnak tenni a vizsgált algoritmusok között. E kutatásban ezt a jelenséget is vizsgáltuk.

4.1.4. 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 szemben 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 érdekelték voltunk abban, hogy mennyire érinti ez a kétfajta figyelemelterelő elem az AlgoRhythmic 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).

4.1.5. Kutatási kérdések

A korábbi kutatások, amelyek az AlgoRhythmic környezetet vizsgálták, ami a tánc-koreográfiákat 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 tánc-koreográfia vizualizációja (realisztikus ábrázolás) köré épült.

A fenti szakirodalmi áttekintés nyomán az alábbi kutatási kérdéseket foglalmaztuk 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 tánc-koreográfiák ábrázolták, mint ha az illusztrálásuk sematikus ábrázolások révén történt volna?

3. Hogyan befolyásolják a realisztikus vizualizációkkal való tanulás hatékonyságát a dekorációs, illetve reprezentációs figyelemelterelő elemek?

4.1.6. Kísérlet

A kísérletet háromszakaszosra terveztük: előteszt, tanulási szakasz, utóteszt. A kísérletre a 2019–2020-as tanév elején került sor, az első héten az előtesztre, a másodikon a tanulási szakaszra és az utótesztre.

A kutatás résztvevői

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

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. táblázata mutatja be. A tanulási szakaszhoz a buborék- és kiválasztó rendezések, AlgoRhythmic környezetbeli sematikus és realisztikus vizualizációit választottuk. Mindkét algoritmus úgy fogható fel, mint a „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.

4.1.7. Eredmények és kiértékelés

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

A két csoport előteszt-teljesítményének az összehasonlításához varianciaanalízist (ANOVA) használtunk. Ami az utóteszteredményeket illeti, ezek vizsgálatához a variancia-kovariancia-analízist (ANCOVA) választottuk. E tesztek használatának egyik előfeltétele, hogy az összehasonlítandó csoportok szórás-homogenitása egyenlő legyen. Ennek ellenőrzése végett Levene-próbát alkalmaztunk, amely minden esetben szignifikanciaszint 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.

1. táblázat. *A Fisher-féle egzakt teszt eredménye*

	Nem		Hány évet tanult programozást?			Tanultad a buborékrendezést?		Tanultad a kiválasztó rendezést?	
	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-féle 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) $p = 0,27 > 0,05$	

Az előteszteredmények 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 válaszai 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 utótesztválaszait a következőképpen kódoltuk (mindkét algoritmus kapcsán a 0–13 intervallumból kaphattak pontokat).

– Az első 4 kérdés válaszaira (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 helyett 15-öt választottak (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.

4.1.8. Absztrakt animációk versus tánckoreográfiák

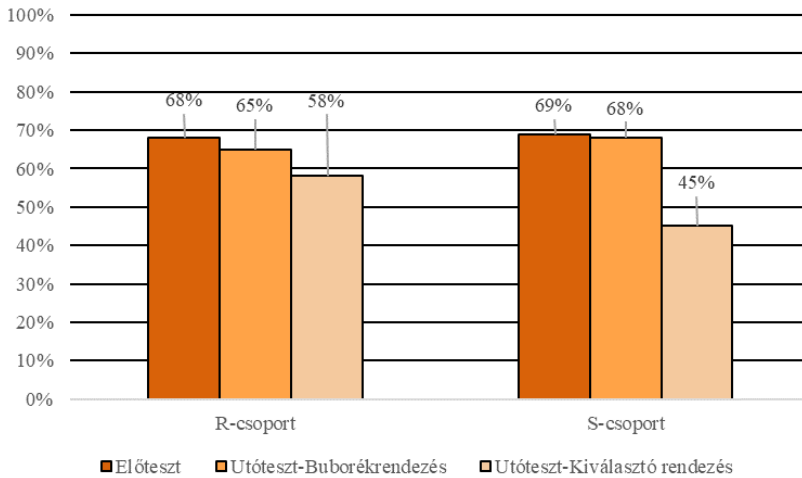
Az utóteszt eredményeinek az összehasonlítására az ANCOVA-tesztet használtuk. A független változó újra a tanmenet típusa volt, a függő változó pedig

az utótesztkérdésekre adott válaszok pontszámai. Az előteszteredményeket mint kovariánst 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 kontrollcsoport (S csoport) ($F(1,113)=9,52$, $p=0,002<0,05$; $\text{partial } \eta^2 = 0,07$) (2. táblázat). Az ANCOVA-teszt is azt mutatta, hogy ezt az eredményt nem befolyásolta az előtesztnél tapasztalt minimális (nem szignifikáns) különbség.

2. táblázat. *Elő- és utóteszteredmények a két csoport szignifikanciakülönbségének mérésére*

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

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%).



19. ábra. *A két csoport előteszt- és utóteszteredményei*

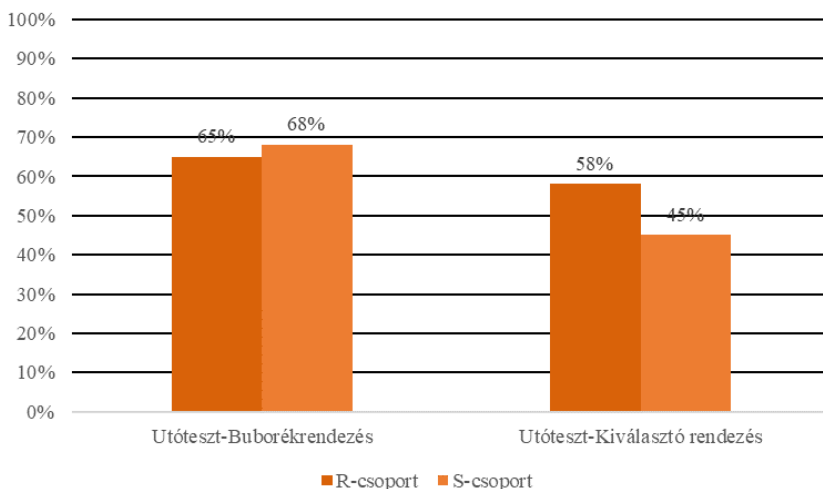
Ez várható is volt, hiszen lényegesen több résztvevő válaszolta azt, hogy már ismerte a buborékrendezést, mint ahányan a kiválasztó rendezést (bár az összes

diák részére, akik tanultak programozást a középiskolában, a tanterv mindkét algoritmust tartalmazta).

Az első kutatási kérdésünkkel kapcsolatban az eredmények azt mutatják, hogy az R csoport tagjai nem teljesítettek alul az S csoporthoz képest (19. 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 realiztikus 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.



20. 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 realiztikus 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 jelentek 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 buborékrendezés 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):

- Az 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):

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

Az 1–4. kérdéseknél (a buborékrendezésre 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 értetődő 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 buborékrendezés-videóban a „hasonlít+csere” műveletpárnak 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 tánc lépésekkel 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 figyelemelterelőkől adódó esetleges hátrányt, addig a reprezentációs figyelemelterelők jelenléte gyengítheti ezt a hatást, és negatívan befolyásolhatja a realisztikus ábrázolások hatékonyságát. Ez egy észszerű 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.

4.1.9. Lányok versus fiúk

A fiúk és lányok utóteszteredményeinek az összehasonlítására kétszemponos varianciaanalízist (ANOVA) használtunk, amelyben a két független változónak a tanmenet típusát és a nemet, függő változónak pedig az utóteszt teljesítményt választottuk. A vizsgálat azt mutatta, hogy egyik algoritmus esetében sem volt összefüggés a két független változó között, más szóval mindkét nem egyformán teljesített, függetlenül attól, hogy milyen tanmenet szerint tekintette meg az algoritmusokat. Ez az eredmény összecseng korábbi tanulmányok következtetéseivel (Ahadi et al. 2017), melyek alapján kijelenthető, hogy nem függ a tanulók nemétől a diákok teljesítménye programozásorientált vagy számítógépesgondolkodás-orientált környezetekben.

4.1.10. Következtetések

Ebben a fejezetben a sematikus, illetve realisztikus dinamikus vizualizációkat vettük nagytű alá, egészen pontosan összehasonlítottuk az AlgoRhythmic környezethez tartozó absztrakt animációkat és realisztikus videókat mint eszközöket a számítógépes algoritmusok oktatásához. Az AlgoRhythmic videók egyedi sajátossága, hogy tánckoreográfiákkal ábrázolnak keresési és rendezési algoritmusokat. Ebből adódik az, hogy hasznosítják az úgynevezett emberimozgás-effektust.

A bemutatott kutatás eredményeiből arra következtethetünk, hogy amennyiben egy realisztikus vizualizáció emberi mozgást alkalmaz, ugyanolyan hatékony tud lenni, sőt akár hatékonyabb is, mint a megfelelő sematikus ábrázolás. Ez azért fontos következtetés, mert korábbi kutatások inkább a sematikus vizualizációk hatékonyságát hangsúlyozták, merthogy kevesebb figyelemelterelő 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 figyelemelterelő 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.

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

4.2. Sematikus és realisztikus algoritmusvizualizációk ismételt megtekintése, illetve ezek kombinálása

4.2.1. 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 információt (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 miként 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 feltétlenül 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 feltétlenül 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, vagy mind szöveges, mind képi formátumban 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 lehet, hogy a szöveges rész segíti a grafikus anyag értelmezését. Ettől eltérően, a többszörös grafikus vizualizációk alkalmazásakor, ugyanolyan típusú ábrázolásokról lévén szó, 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 miként 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).

4.2.2. 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 AlgoRhythmic környezet révén történne (21. ábra).



21. ábra. Statikus vs. dinamikus ábrázolásmód

4.2.3. Kutatási kérdések

A szakirodalmi áttekintés nyomán az alábbi kutatási kérdéseket fogalmaztuk meg:

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

4.2.4. Kísérlet

Erre a kísérletre a 2018–2019-es tanév elején került sor, és három szakaszt foglalt magába: az előtesztet, a tanulási szakaszt és az utótesztet.

A kutatás résztvevői

Az előtesztnek szánt felmérésre meghívtunk minden elsőéves reál szakos 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 részt vevő középiskolás tantervében.

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

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 pszeudokódja). 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 realisztikus vizualizáció az algoritmus tánc Koreográfiája volt. Mivel az AlgoRhythmic számítógépes animációk akkor még fejlesztés alatt álltak, sematikus ábrázolásként egy jó minőségű, nagy nézettséggel rendelkező YouTube-videót választottunk (New Media, 2013).

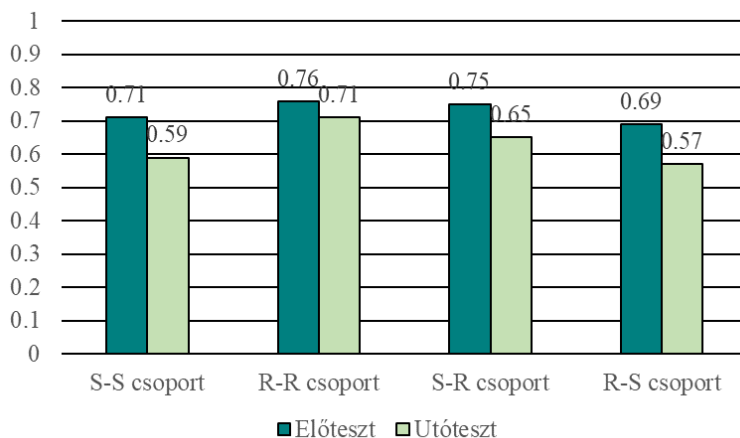
Az utóteszt 14 kérdése azt szándékozott felmérni, hogy a résztvevők mennyire értették meg az algoritmust, azaz el tudják-e képzelni a működését egy újabb véletlen számsorozaton, illetve növekvő („best-case”) és csökkenő („worst-case”) számsorozatokon.

4.2.5. Eredmények és kiértékelés

A négy csoport előteszteredményeinek összehasonlításához egyszempontos varianciaanalízist (ANOVA) használtunk. A független változó a tanmenet típusa volt, a függő változó pedig a résztvevők előteszt-teljesítménye (22. ábra). Az elemzés arra mutatott rá, hogy a négy csoport között nincs szignifikáns eltérés ($F(3,79) = 0,49$), azaz statisztikailag egyenértékűnek tekinthetők.

3. táblázat. *A csoportonkénti eredmények átlagai és zárójelben a megfelelő szórásértékek*

	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)

**22. ábra.** *A csoportonkénti eredmények átlagainak ábrázolása*

Az utóteszt-teljesítmények kiértékeléséhez variancia-kovariancia-elemzést (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 tánc-koreográfia-vizualizációt (emberi mozgást alkalmazó realiztikus ábrázolást). Ez arra enged következtetni, hogy nem jelenthető ki, hogy az emberimozgás-effektust hasznosító ábrázolások, bármely kombinációban jelenjenek is meg, hatékonyabb tanulást eredményeznek. Az először a realiztikus (tánc-koreográfia), majd a sematikus (animáció) sorrend szembement a multimédiával támogatott oktatás azon alapelvével, 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 újdonságszámba ment a tánc-koreográfiába kódolt algoritmus gondolata.

Másfelől viszont az eredmények arra mutatnak, hogy azok a diákok, akik kétszer látták a realiztikus 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 realiztikus 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 összpontosítottunk, hogy sematikus versus realiztikus, ellentmond a Scheiter és társai (2009) következtetésének (abban a kutatásban a realiztikus 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 komplementaritás 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 a 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 tanmenettípusnak 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 diákokat 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 megértsék 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.

4.2.6. Következtetések

Ebben a kutatásban azt vizsgáltuk, hogy amennyiben ugyanarról az algoritmusról két ábrázolással is rendelkezünk, sematikusval és realisztikusval 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 sematikusval látták kétszer, nemcsak 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 beigazódott, hogy annak is meg lehet 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 aztán 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).

4.3. Az interaktivitás szerepe online algoritmusvizualizációs környezetekben

4.3.1. Szakirodalmi áttekintés

A dinamikus vizualizációk hatékonyságát vizsgáló szakirodalom egy szelete 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. A 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

szerkesztő 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 tánc-koreográfiával 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 a bemeneten, amelyen az animáció is zajlott, másodszor véletlen bemeneti adatokra, 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 humánorientált 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, tekintsük „ $\frac{1}{2}$ 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 interaktivitástípusnak 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.

4.3.2. Kutatási kérdések

Az előző fejezetben már kifejtettük, hogy haszná lehet egy vizualizáció kétszeri megtekintésének. 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 környezetben?

4.3.3. Kísérlet

Ez a kutatás egyfajta folytatása az előző fejezetben bemutatott tanulmánynak. Arra a 2018–2019-es tanév elején, erre pedig a 2019–2020-as tanév elején került sor. A 2018–2019-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.

A kutatás résztvevői

A kísérletben 134 elsőéves reál szakos 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 előteszteredmények tekintetében (4. táblázat. Résztvevők).

4. táblázat. *Résztevők átlagai csoportokra bontva*

	Létszám	Lányok aránya (%-ban)	Középiskolában tanult programozást (évek átlaga)	Előteszt- teljesítmény (%-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

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 utótesztkérdéssort alkalmaztunk, mint az előző fejezetben bemutatott vizsgálat esetében.

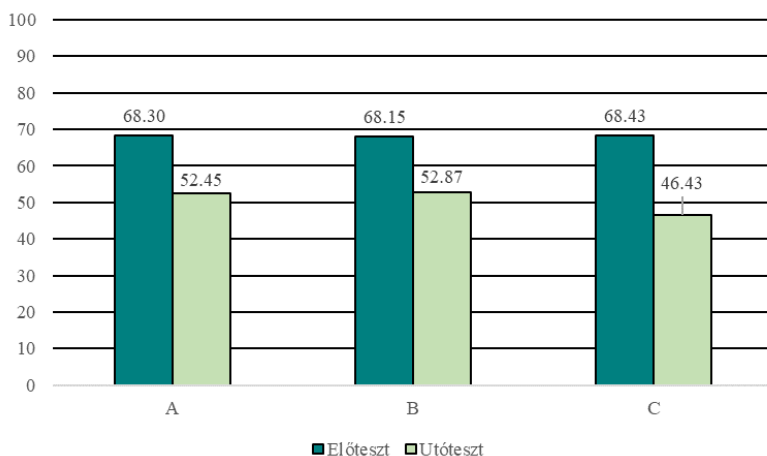
4.3.4. Eredmények és kiértékelés

Az előtesztválaszokat 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 utóteszteredmények vizsgálatánál variancia-kovariancia-elemzést (ANCOVA) használtunk. A független változó ugyancsak az alkalmazott interaktivitási szint volt, a függő változó viszont az utóteszt-teljesítmény. Kovariánsként kezeltük az előtesztpontszámokat. 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 tanmenettípusnak megvannak az előnyei, illetve hátrányai.

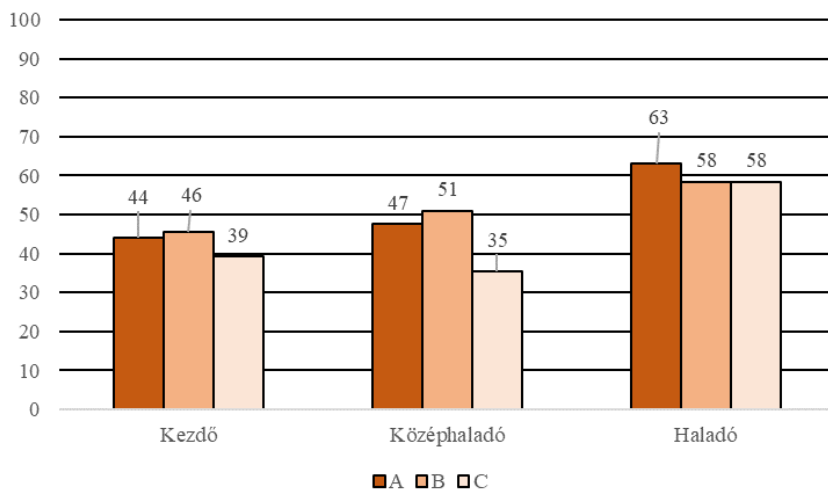
Mivel az előtesztelés során összességében nem találtunk szignifikáns különbséget a különböző csoportok között (23. ábra), megvizsgáltuk a tanulók utóteszten elért teljesítményét az előzetes programozási tapasztalatok függvényében is.

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).



23. ábra. *Előteszt és utóteszt eredményei csoportonként*

A haladók (3 vagy 4 évet tanultak programozást) eredményeit nem befolyásolta a tanulás interaktivitási szintje (T3.1. tézis) (24. ábra).

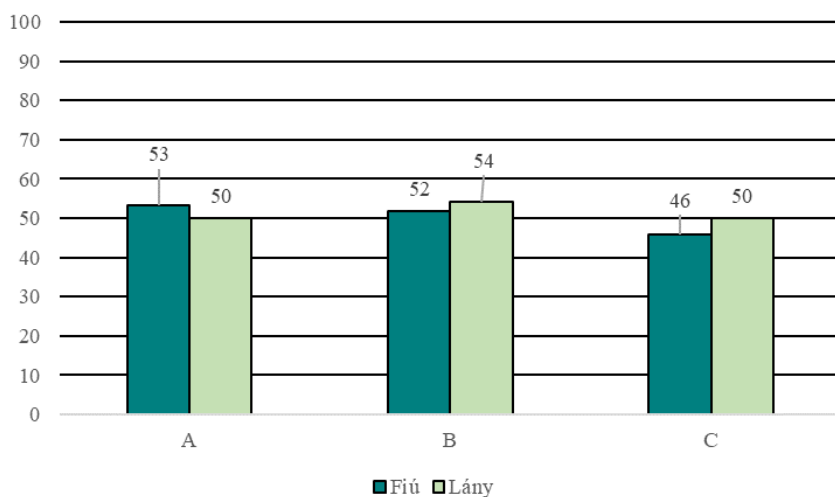


24. ábra. *A 3 programozási szinten levő diákok eredményei az utóteszt során, csoportonként*

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 fölé kerekedtek (T3.2. tézis) (25., 26. ábra).



25. ábra. Az előteszt és utóteszt eredményei összesítve nemenként

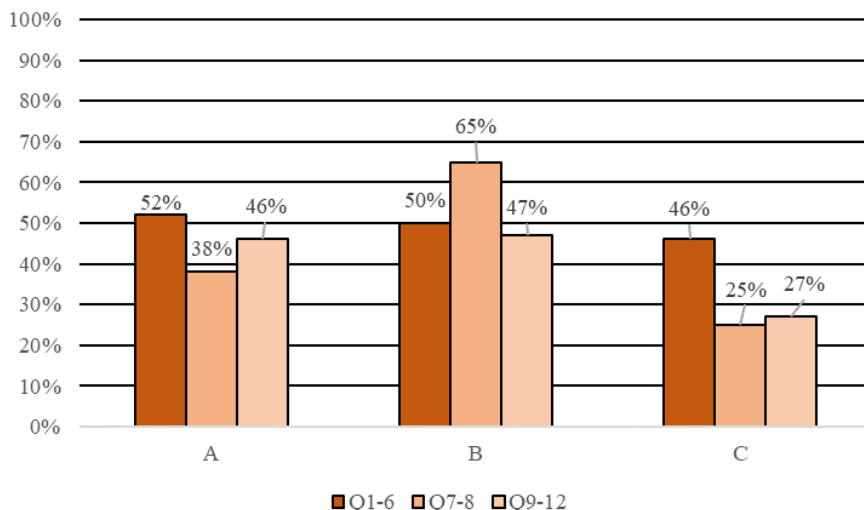


26. ábra. Az előteszt és utóteszt eredményei nemenként csoportokra lebontva

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

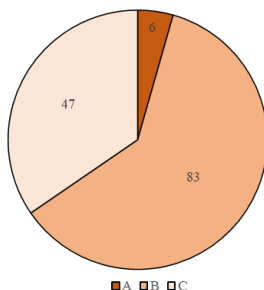
– A 7–8. kérdések az algoritmus egy-egy adott pontján történő művelethez kapcsolódtak. Itt azok a tanulók értek el szignifikánsan jó eredményeket, akik „ $\frac{1}{2}$ interaktivitással” tanultak, vagyis előre nem ismert időpillanatokban kellett bekapcsolódjanak a vizualizációba, és ez meghozta gyümölcsét. 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) (27. ábra).



27. 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 számokra a leghatékonyabbnak (függetlenül attól, hogy melyik tanfolyamcsoportba tartoztál)?”, a diákok szignifikánsan a 1/2 interaktivitással való tanulást választották (28. ábra).



Magyarázat: A: 0–, B: 1/2, illetve C: 1–interaktivitással

28. ábra. A diákok véleménye a leghatékonyabb tanulási módszerről

4.3.5. Következtetések

Ebben a kutatásban azt tanulmányoztuk, hogy miként 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 szintéziskérdések megválaszolásában is lemaradtak (T3.3. tézis).

4.4. Tanári kérdéssorok algoritmusvizualizációs környezetekben

4.4.1. Szakirodalmi áttekintés

A saját tapasztalataink és kollégáink visszajelzései is azt hangsúlyozták, hogy a diákok, amennyiben magukban használják az AlgoRhythmics-környezetet, csak részlegesen tudják kiaknázni a potenciált, amit magában rejt. Például azok a definíciók, amelyek megpróbálják körülírni a számítógépes gondolkodás fogalmát, hangsúlyozzák, hogy fontos, hogy a tanulók a tanulmányozott algoritmusok hatékonyságvizsgálatában 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 diákot, hogy egy adott irányba gondolkodjon, és olyan tartalmakat is érzékeljen, amikre magától nem igazán jönne rá (Guzdial–Tew 2006).

Egy hatékony módja annak, hogy a tanár úgy serkentse a diákot, 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 felé, 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, 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 kifejezőnek kell lennie.

Az előző fejezetekben bemutatott kutatások mindenikében megjelentek a vizualizációhoz társított, algoritmusbonyolultsági szempontokra is kitérő kérdéssorozatok. Ezekre minden esetben úgy tekintettünk, mint utótesztkérdésekre, 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 a tanár irányított kérdés-felelet megbeszéléssel is.

4.4.2. Kutatási kérdések

E kutatás magvát képező kísérletet háromszakaszosra terveztük. Az első szakaszban a résztvevők megtekintették három algoritmus AlgoRhythmicsvizualizációját, a másodikban egy csoport szintű, 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 szakaszban 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 algoritmusbonyolultsági 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 válaszáira, illetve ezek miértjeire.

Az alábbi kutatási kérdéseket foglalmaztuk meg:

1. Elég kifejezőek-e az AlgoRhythmics-vizualizációk ahhoz, hogy egy gondosan összeállított kérdéssorozat rá tudjon vezetni programozói előismeretekkel nem rendelkező elsőéves hallgatókat algoritmusbonyolultsági tartalmakra?

2. Hozzá tud-e járulni a második szakaszbeli kérdés-feleletes megbeszélés ahhoz, hogy a harmadik szakaszban jobban teljesítsenek a résztvevők, mint ahogy az első szakaszban teljesítettek?

4.4.3. Kísérlet

Erre a kutatásra a 2017–2018-as tanév elején került sor.

A kutatás résztvevői

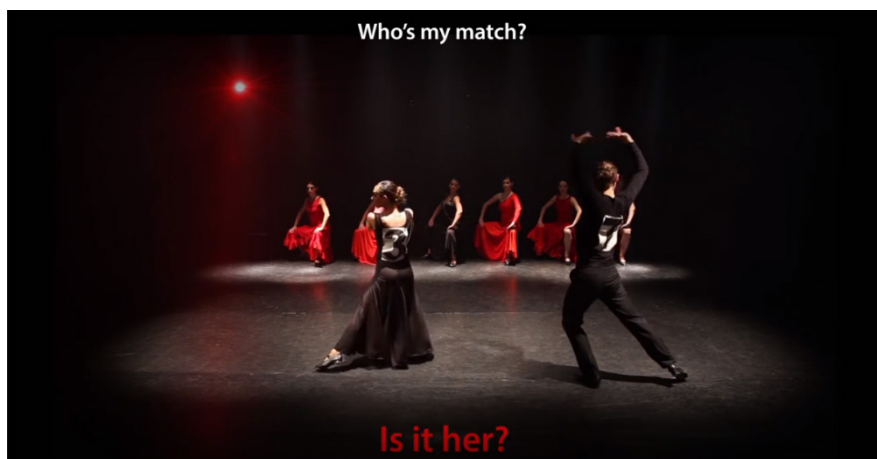
A kísérletbe, ez esetben is, minden elsőéves, reál szakon tanuló hallgatót bevontunk (181 diák, 13% lány). A résztvevőket annak alapján soroltuk csoportokba, hogy hány évig tanultak programozást a középiskolai tanulmányaik alatt: A csoport (0 évet tanultak, 27%); B csoport (1 vagy 2 évet tanultak, 25%); 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 algoritmusbonyolultság 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 kontrollcsoportnak. Kíváncsiak voltunk rá, hogy megvan-e az AlgoRhythmics-környezetben a potenciál ahhoz, hogy az A csoport diákjait felzárkóztassa a B csoportéhoz. A C csoport ebben az elemzésben egyfajta másodlagos kontrollcsoportként lett bevonva.

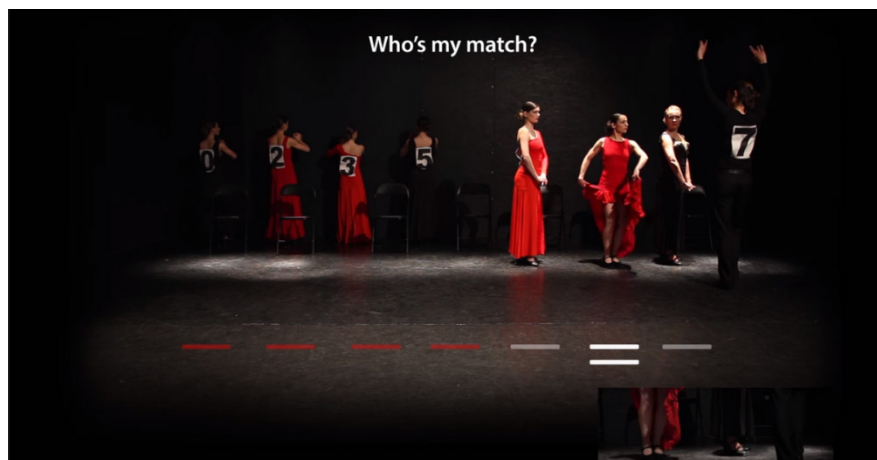
A kísérlet második és harmadik szakaszában csak az A csoport tagjai vettek részt. Arra számítottunk, hogy miután e résztvevők elmélyültek az első szakasz kérdéseiben, és aktívan részt vettek a második szakasz keretében zajló megbeszélésen, a harmadik szakaszban szignifikánsan jobban fognak teljesíteni,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

A kutatás megvalósításának eszközei

Az első szakasz a lineáris és bináris keresések, valamint a buborékredezés vizualizációira épült (7, illetve 10 elemű véletlen számsorozatokon illusztrálva az algoritmusok működését). Mivel a keresési algoritmusok viszonylag egyszerűek, ezek csak az AlgoRhythmics tánc Koreográfiákkal voltak illusztrálva (29., 30. ábra).



29. ábra. Lineáris keresés flamencotáncsal



30. ábra. Bináris keresés flamencotáncsal

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 kérdéssorozatokat. 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 (használt és csere). Kifejezetten ösztönöztük őket arra, hogy csak a második megtekintés alatt összpontosítsanak 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 szcenárió 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.

4.4.4. 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 mind I., mind II. típusú ismerete is volt (középiskolai tanulmányaikból adódóan). Bár a kérdésekre adott válaszok a II. típusú ismeretet tükrözték, ez a 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 Melléklet 14. táblázata 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 szóban forgó algoritmus. A bináris keresés esetében, amelynek logaritmikus az időbonyolultsága, a csoport átlagteljesítménye csak 34% volt. Észszerű 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 előzőleg mindhárom algoritmust tanulták, 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 szintéziskérdés 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 legkevésbé intuitív algoritmus. Nyilván a C csoport szignifikánsan jobban teljesített, mint az A 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, 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), nyilvánvaló előnyt jelentett számukra az A csoporthoz viszonyítva. 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 mindezek 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 eredményt, 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 csoporthoz képest a kevésbé intuitív algoritmus esetében, illetve hogy mind az A, mind a B csoport szignifikánsan lemaradt a C csoporthoz 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, ugyan-csak erre a következtetésre jutottunk.

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

	Legjobb eset (hasonlítás)	Legjobb eset (csere)	Legrosszabb eset (hasonlítás)	Legrosszabb eset (csere)
Első szakasz (buborékrendezés)	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%

Amint az 5. táblázat 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 a 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 figyelemre méltó 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 a 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).

4.4.5. 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 szakaszokban 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 AlgoRhythmics-videók elég kifejezőek 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.

4.5. Számítógépes gondolkodás vizsgálata különböző korosztályú tanulóknál

4.5.1. 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 érdekében 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 e képesség fejlődéséhez (Evans–Simkin 1989). Ezzel összhangban számos kutatás, amelyet elsőséves egyetemi hallgatókkal végeztek, úgy következtet, hogy a diákok nem rendelkeznek azokkal a képességekkel, amelyekre a programozásoktatásnak 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 kutatásösszefoglaló, 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).

4.5.2. Az AlgoRhythmics-ötletre épülő számítógépes gondolkodási 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, amelynek háttérül egy számítógépes algoritmus (lineáris keresés) szolgált. A feladat megszövegezésekor odafigyeltünk arra, hogy a 3. osztályos tanulók számára is

érthető és érdekes legyen. Az alapelvet, amelyet figyelembe vettünk, Comenius a következőképpen fogalmazta meg: „A gyermekekkel megtaníthatók a felnőttek 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 sorra felkérje egy-egy táncra a lányokat.

Ahogy a bevezetőben már utaltunk rá, főleg olyan algoritmusok tanulmányozása járul hozzá a számítógépes gondolkodás fejlesztéséhez, amelyek háttérében 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, hibakeresés, 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 találtatik. Az algoritmus átlátása azt feltételezte, hogy a tanulóknak el kellett tudniuk vonatkoztatni (absztrakció) olyan elemektől, mint fiú, lányok, tánc stb. A csatolt kérdések (4.5.3. 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.

4.5.3. Kutatási kérdések

A jelen kutatáshoz az alábbi kutatási kérdéseket fogalmaztuk meg:

1. Észlelhető-e fejlődés abban (a jelenlegi curriculum tantárgyai hozadéka-ként), 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?

4.5.4. 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 szakaszban igyekeztünk 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 tánc-koreográfia-ábrázolását. A harmadik szakaszban a résztvevők olyan kérdésekkel szembesültek, amelyek feltárhatták, hogy mennyire értették meg a vizualizált algoritmust.

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.

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

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

Azért kezdtük a 3. osztállyal, mert a 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 korosztály szerinti fejlődés. E felosztás harmonizál a CSTA (2017) (Computer Science Teachers Association) 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.

A kutatás megvalósításának eszközei

A kísérletre a tanulók által megszokott, saját oktatási környezetükben, az iskolájuk számítógépekkel felszerelt informatikaboratóriumában 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 AlgoRythmics-környezet lineáris keresést ábrázoló videóját vetítettük le a diákoknak, amely flamencotáncsal illusztrálja a szóban forgó algoritmust.

4.5.5. 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 38. 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 40. 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. Meglepően 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 érthettek el. A jó válaszokért járó pontok évfolyamonkénti átlagait (százalékban kifejezve) és a megfelelő szórásértékeket 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 vettünk figyelembe (7. táblázat).

7. táblázat. *A kutatás kérdéseire adott válaszok átlaga és zárójelben a megfelelő szórásérték*

	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)

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 válaszait t-teszt, valamint egy- és kétszemponos varianciaanalízis (ANOVA) segítségével dolgoztuk fel. Levene-próbával ellenőriztük ezek alkalmazhatóságát ($p=0,07>0,05$).

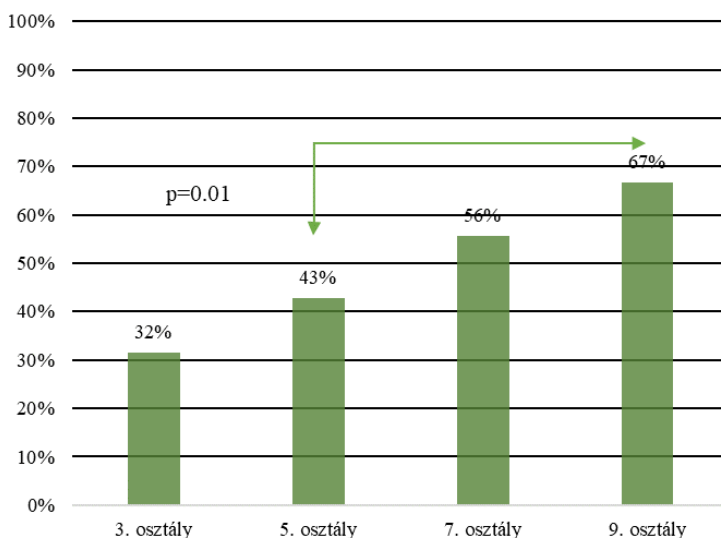
4.5.6. Korosztályonkénti elemzés

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 éreznek rá olyan tanulók, akik programozást nem tanultak, egy számítógépes algoritmusra. A 31. 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 kontingenciátáblákba 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).

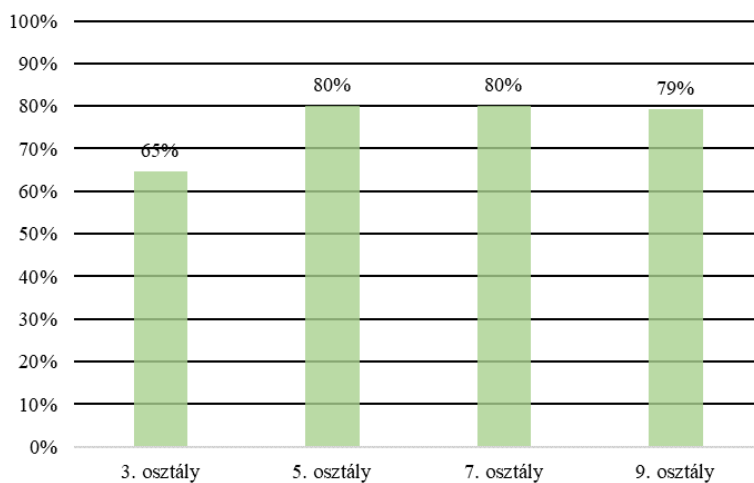
8. táblázat. Az első kérdésre adott helyes és helytelen válaszok száma osztályonként

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

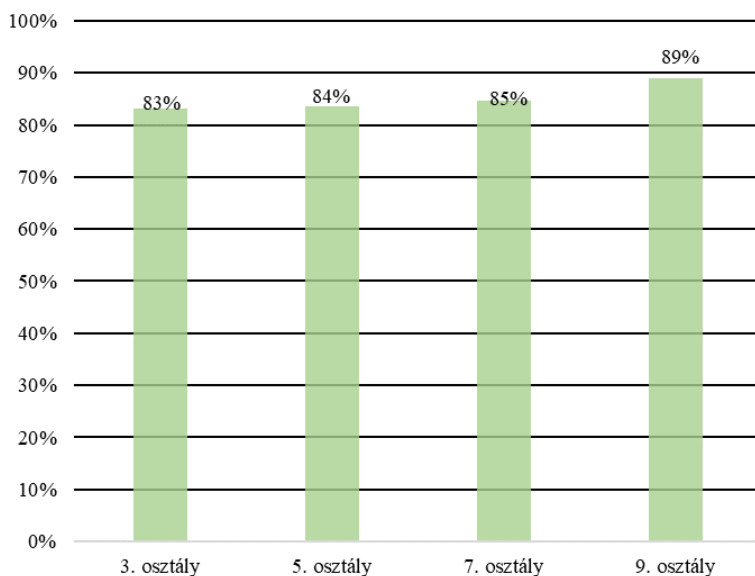


31. ábra. A kísérlet első szakaszának eredményei korosztály szerint (mindkét iskola)

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 eredményekre 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).



32. ábra. A kísérlet harmadik szakaszának eredményei korosztály szerint (mindkét iskola)



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

A kísérlet harmadik szakaszában feltett kérdéssorozatra érkezett válaszokat egyszempontos varianciaanalízissel (ANOVA) elemeztük. Az eredmények azt mutatták (32. 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 $(3, -1, -1, -1)$ kontrasztértékeket 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 ($p = 0,47$) (33. ábra).

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).

4.5.7. 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 részt vevő 3M osztályos tanulók a zene szakon, az 5M, 7M, 9M osztályos tanulók a rajz szakon 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 kisérettségi 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.

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

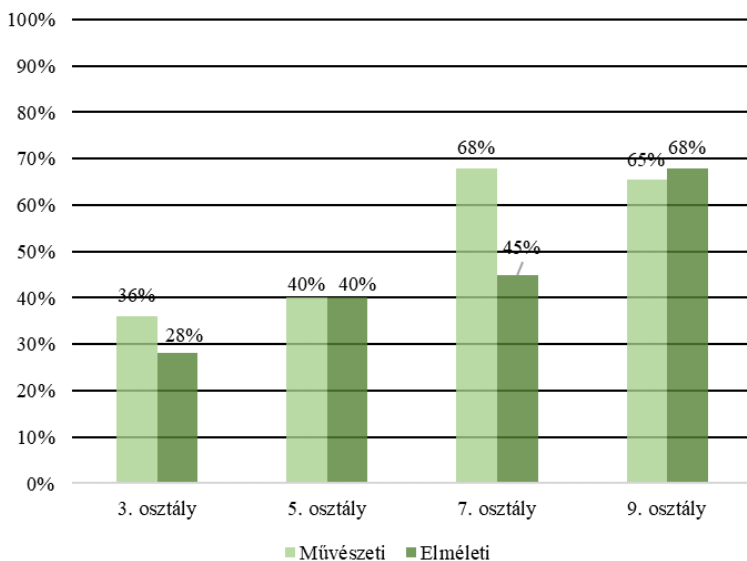
Míndezen alapján kijelenthető, hogy az iskolák szerinti elemzéshez a legrelevánsabb összehasonlítások a 3M vs. 3E, illetve 7M vs. 7E szembeállítások, hisz ezen osztályok tanulói legalább két évig folytonosan az adott szakirány tanterve alapján tanultak.

9. táblázat. *Adott iskolában tanult évek száma szakterületenként*

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 tanulmányukat
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 beállítottaságú diákok
9E	–	8 év (1–8. osztály)	–	Jó képességű diákok érkeznek mindenfelől

A 34. ábra a K_1 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).



34. 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).

4.5.8. 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 K_1 kérdés kapcsán mindkét nem teljesítménye 49%-os volt. A harmadik szakasz kérdés-sorozatán 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, kétszemponos 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 eredményekre jutottak, például Lau és Yuen (2009) kutatása, akik 270, 14–19 év közötti, középiskolás tanulón mértek, vagy 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 összehasonlításához kétmintás t-próbát használtunk, ahol a független változó a tanulók neme, a függő változó pedig a megfelelő kérdéshez tartozó pontszám volt (zárójelben az átlagokat és a megfelelő szórásértékeket 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 önbizalomhiányát 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 válaszai mutattak magasabb értékeket.

4.5.9. 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 hozadékaként létezik egyfajta növekedés, de ez viszonylag lassú. Kétévenkénti eltolásban nem mutatható ki, viszont négyévenkéntiben igen. Egy érdekes észrevételünk, hogy úgy tűnik, a művészeti 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 (35. ábra).



35. ábra. A kutatási kérdésekre adott válaszok feldolgozása nemenként

5. A KUTATÁSOK ÖSSZEGZÉSE

A bemutatott öt kutatásban különböző korosztályú tanulók számítógépes gondolkodása fejlesztésének lehetőségeire összpontosítottunk interaktív, több érzékszerves, elektronikus oktatási környezetekben. Egészen pontosan azt vizsgáltuk, hogy miként járulhatnak ehhez hozzá az AlgoRhythmics-környezetbe ágyazott algoritmusvizualizációk.

5.1. Következtetések

A kutatócsoport már a kezdetektől tudatában volt annak, hogy a tánc-koreográfia-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 elkészültek az újabb algoritmus-illusztrációk, úgy vált nyilvánvalóvá, hogy az AlgoRhythmics webes környezet is megújításra szorul. Néhány fejlesztés, amelyben kulcsfontosságúnak bizonyultak, 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 tátongó szakadék áthidalásához; tegye lehetővé a környezet a felhasználók teljesítményének mérését.

A 2016 előtt megvalósított kutatások többnyire az AlgoRhythmics-környezet több érzékszerves jellegét vizsgálták. A kapcsolódó publikációk címe is ezt tükrözi: *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. Ezt követően új szempontok kerültek előtérbe.

Új nézőpontból tekintve a kétfajta AlgoRhythmics-ábrázolásra, világossá vált, hogy a tánc-koreográfiák a realisztikus, a számítógépes animációk pedig a sematikus dinamikus ábrázolások kategóriájához köthetők. Az egyik kutatási vonal, amelyet követtünk, tehát az volt, hogy e szempontból tanulmányozzuk az AlgoRhythmics-vizualizációkat. A szakirodalmi áttekintés után világossá vált, hogy a tánc-koreográfiák 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, két-féle figyelemelterelő elemet különböztettünk meg az AlgoRhythmics-videókban:

reprezentációt, amikor a művészi jegyek árnyékot vetnek az algoritmus kulcsműveleteinek ábrázolására, és dekorációs, amelyeknek csak díszítő és motiváló szerepük van. Mivel az AlgoRythmics-környezetben mindkét típusú ábrázolás elérhető (sematikus és realisztikus), ezért felmerült az a kérdés is, hogy ezek különböző módokon való kombinálása hatékonyabb tanulást eredményez-e. Az 1. és 2. kutatás révén, amelyeket ebben a témakörben végeztünk, az alábbi téziseket sikerült bizonyítani:

[T1.1] Azok a realisztikus algoritmusvizualizációk (mint például az AlgoRythmics 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, mint ha sematikus ábrázolások lettek voltak alkalmazva.

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

[T2.1] Amennyiben ismételt megtekintést alkalmazunk, az emberimozgás-effektust hasznosító realisztikus algoritmusvizualizációk (mint például az AlgoRythmics-videók) hatékonyabbak, mint a megfelelő sematikus ábrázolások (például absztrakt számítógépes animációk).

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

Mivel a könyv alapjának egyik főpontja az interaktivitás szerepének a vizsgálata volt, ezért a 3. kutatásban azt vizsgáltuk, hogy nulla (a vizualizáció zavar-talan megfigyelése), fél (előre nem ismert időpillanatokban megáll a vizualizáció, és a tanuló meg kell mondja a következő lépést), vagy teljes interaktivitás (a tanulóknak le kell vezényelnie a teljes algoritmust) vezet a tanulmányozott algoritmus stratégiájának jobb megértéséhez. A szakirodalommal összhangban az igazolódott be, 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 algoritmusvizualizáció 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.

Saját oktatói tapasztalatokból és más kollégák visszajelzéseiből is az derült ki, hogy amennyiben a diákok magukban használják az AlgoRythmics-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 a vizsgálatok arra is kitértek, hogy amennyiben kifejező algoritmusvizualizációk mellé jól átgondolt tanári kérdéssorokat társítunk, képesek-e a tanulók akár algoritmusbonyolultsági szempontokat is kihámozni az ábrázolásokból. A 4. 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.

Az előbbi mérések mind reál szakos elsőéves egyetemi hallgatókon történtek. Az egyik legnagyobb kihívást az jelentette számunkra, hogy kiterjesszük a vizsgálatot elemi, gimnazista és középiskolás tanulókra is. E kutatás érdekes következtetésekhöz 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 expliciten a számítógépes gondolkodás fejlesztésére, akkor a többi tantárgy 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.

6. KÖVETKEZTETÉSEK

Az emberközpontú algoritmusvizualizációk és az interaktív, több érzékszerves elektronikus oktatási környezetek együttesen kulcsfontosságú szerepet játszanak a számítógépes gondolkodás fejlesztésében. Az algoritmusvizualizációk lenyűgöző módon mutatják be az algoritmusok működését, amelyek segítenek a diákoknak könnyebben megérteni és elsajátítani ezek működését. Az interaktív elektronikus oktatási környezet lehetőséget teremt a diákoknak, hogy saját tempójukban és a számukra legmegfelelőbb módszer szerint tanuljanak, így elősegítve a mélyebb, alaposabb megértést és a kreatív problémamegoldó képesség fejlődését is.

A számítógépes gondolkodás kritikus képesség mai, modern világunkban. A folyamatosan megjelenő innovatív oktatási megközelítések egyedülálló módon segítik a diákok fejlődését ezen a területen. Az interaktív oktatási környezetek lehetővé teszik az algoritmusok gyakorlatban való alkalmazását, interaktív játékokat és kihívásokat biztosít, amelyek felkeltik a felhasználók érdeklődését, és motiválják őket a tanulásra.

Ezek a fejlesztések nemcsak az algoritmusok és a programozás megértését segítik elő, hanem a kritikus gondolkodást, az elemző és a problémamegoldó képességet is fejlesztik. Az interaktív, több érzékszerves elektronikus oktatási környezet elősegíti a diákok kreativitását és egyedi tanulási stílusát, miközben felkészíti őket a digitális kor kihívásaira.

Összességében ezek az újítások forradalmasítják az oktatást, és lehetővé teszik a diákok számára, hogy élvezetes, interaktív és hatékony módon fejlesszék számítógépes gondolkodásukat. Az interaktív, több érzékszerves elektronikus oktatási környezet egy újabb lépés a digitális oktatás jövőjének irányába, amely lehetővé teszi, hogy a diákok fejlett számítástechnikai készségekkel lépjenek be a világ egyre fejlődő és digitalizált társadalmába.

Az ilyen és ehhez hasonló módszerek népszerűségének következtében elengedhetetlen lépést tartani ezek fejlődésével. A digitális kor diákjainak érdeklődését egyre nehezebb lesz fenntartani, és egyre nagyobb kihívást fog jelenteni az is, hogy motivációjukat fenntartsuk. Az emberközpontú vizualizációk és az interaktív környezetek az oktatás jövőjének elengedhetetlen mérföldkövei, éppen ezért folyamatos továbbfejlesztésük is nélkülözhetetlen. Jövőbeli terveink között éppen ezért kiemelt szerepet kap további vizualizációk kidolgozása, mely elsősorban az emberi mozgásra fókuszál, és kihasználja azt az erősségét, hogy az ember mint társadalmi lény azonosulni tud, sőt magáénak tekinti az ilyen jellegű vizualizációkat. Továbbá kiemelt figyelmet szeretnénk fordítani a továbbiakban arra is, hogy minél jobb felhasználói élményt nyújtson az AlgoRhythmic oktatói környezet és a hozzá tartozó videók repertoárja, amelyet szeretnénk minél több helyen népszerűsíteni.

7. MELLÉKLETEK

7.1. A kutatások menete

7.1.1. Első kutatás menete

Az előtesztre az egyetem nagy amfiteátrumában került sor. A hallgatók, a Socrative alkalmazást használva, mobilkészülékek 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 Buborékrendezés algoritmus alapján melyik az első menet első/második/harmadik/negyedik művelete: $\text{Hasonlít}(a[\dots], a[\dots]) / \text{Csere}(a[\dots], a[\dots])$? 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: $\text{Hasonlít}(a[\dots], a[\dots]) / \text{Csere}(a[\dots], a[\dots])$? Írd be a helyes választ szóköz nélkül, a pontokat helyettesítsd a megfelelő számokkal.

– 9. kérdés: Adott a következő számsorozat: 12 7 15 5 6 4 8. A számsorozat elemeit az a tömbben tároljuk és 0-tól sorszámozzuk. A buborékrendezés algoritmus alapján melyik a második menet első művelete: $\text{Hasonlít}(a[\dots], a[\dots]) /$

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

– 10. 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: $\text{Hasonlít}(a[\dots], a[\dots]) / \text{Csere}(a[\dots], a[\dots])$? 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 buborékrendezés 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?

10. táblázat. *A kérdőív szerkezete (Sematikus versus emberi mozgást alkalmazó realisztikus algoritmusvizualizáció)*

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

Kérdőív részei	Tartalmi egységek	Kérdések száma
Blokk	Buborékrendezés (bonyolultsági kérdések és általánosítás – ismeretlen az elemek száma)	4
Blokk	Kiválasztó rendezés (bonyolultsági kérdések és általánosítás – ismeretlen az elemek száma)	4

8.1.2. Második kutatás menete

Az előtesztre egy amfiteátrumban került sor, az összes résztvevő számára egyszerre. A két algoritmus pseudokódját 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 érhettek el):

– 1–2. kérdések: Tekintsük az 5 3 9 1 7 ötelemű számsorozatot. Hány összehasonlítás/csere 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 mindenki a megfelelő tanmenet szerint tekinthette meg a Shell-rendezési algoritmust (36. ábra, 37. ábra). 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.

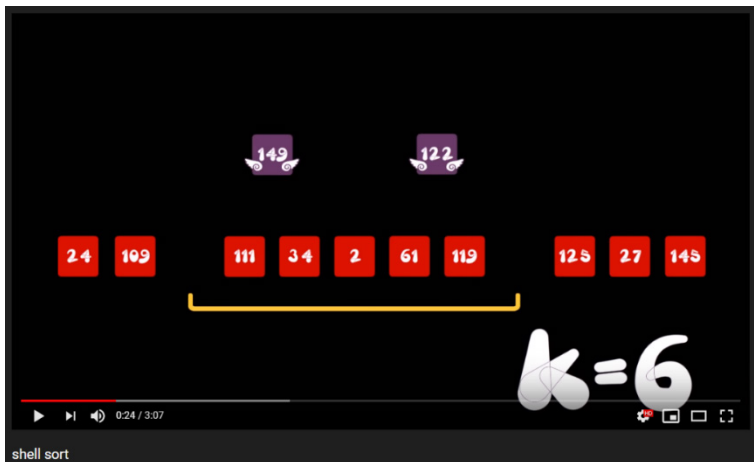
– R–S csoport: először a realisztikus, majd a sematikus vizualizációt tekintette meg, egymást követően.

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 tánckoreográfia 10 elemű számsorozaton, az animáció pedig 12 elemű számsorozaton illusztrálta az algoritmust, ezért az utótesztkérdések egy 7 elemű számsorozatra vonatkoztak, annak érdekében, hogy egyik csoportot se részesítsük előnyben. A számsorozatot, amelyre az alábbi kérdések vonatkoztak,

az x tömb tárolta ($x[1...7]=[1, 19, 7, 8, 12, 11, 9]$), 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 ($x[?], x[?]$))!



36. ábra. A Shell-rendezés realizztikus ábrázolása



37. ábra. A Shell-rendezés sematikus ábrázolása

– 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 hételemű sorozat esetén hány hasonlítás/csere történik?

7.1.3. Harmadik kutatás menete

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

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: buborékredezés	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

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 $x[0...6]=\{1, 19, 7, 8, 12, 11, 9\}$ tömbre vonatkoznak, hogy az algoritmus növekvő sorrendbe rendez, 3 és 1 növekményértékekkel dolgozik, és hogy művelet alatt hasonlítást vagy csere-t értünk.

- 1–3. kérdések: Mely elempárookra vonatkozik az első/második/ harmadik „Művelet ($x[?]$, $x[?]$)”?
- 4–6. kérdések: Az első/második/harmadik művelet: hasonlít vagy csere?
- 7. kérdés: Az $x[3]$ és $x[6]$ elemek összehasonlítása után melyik a következő elempár, amely összehasonlításra kerül?
- 8. 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 hételemtű sorozat esetén hány összehasonlítás/csere történik?
- 13. kérdés: Melyik módszer (interaktivitási szint) tűnik számodra a leghatékonyabbnak (függetlenül attól, hogy melyik csoportba tartoztál)?

7.1.4. Negyedik kutatás menete

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

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 haté- konyságuk alapján	1

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 tánckoreográfiát, a rendezés esetén először a tánckoreográfiát, majd az animációt); 2. válaszoltak a megfelelő kérdéssorozatra (papíron). Az alábbiakban egy Qx.y.z kód úgy értelmezendő, hogy az x szakaszban az y algoritmushoz 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?
- (Q1.1.4–6) 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 sorrendben 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 buborékstratégia 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 szemináriumi termekben zajlott: a résztvevők megkapták a kiválasztó rendezésre vonatkozó kérdéssorozatot, megtekintették az algoritmusnak először a tánckoreográfia-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 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.

13. táblázat. *A kutatás lépései*

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

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)

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

7.1.5. 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? (az 1-es azt jelenti, hogy nagyon nem szereted, a 10-es azt jelenti, hogy nagyon szereted)
- $K_{0.2}$ kérdés: Mennyire szereted a táncot és a zenét? (az 1-es azt jelenti, hogy nagyon nem szereted, a 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 (38. 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 táncokoreográfiát táncolja. A tanulónak azonosulniuk kellett a táncos fiúval, majd a kép tanulmányozása után a tanulók az alábbi kérdésre kellett választ adjanak:

- K_1 kérdés: Hány lánnyal kell táncolnia a fiúnak ahhoz, hogy megtalálja a párját?



38. ábra. A párját kereső fiú és a hételemű lányoszorozat

Ezt követte a második szakasz, amely a lineáris keresést ábrázoló videó megtekintéséből állt (39. ábra). Ezután felkértük a résztvevőket, hogy válaszoljanak arra a kérdésre, hogy:

- K_2 kérdés: Mennyire értetted meg a videón látott fiú alkalmazta keresési stratégiát? (1: egyáltalán nem; 10: teljes mértékben)



39. 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 (40. ábra). A csatolt kérdés is hasonló volt:

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



40. ábra. Lineáris keresés, tízelemű sorozaton

Ahhoz, hogy felmérjük, hogy a résztvevők mennyire mélyen értették meg a vizualizált algoritmust, a következő algoritmusbonyolultsági szempontokra kérdeztünk rá. A következő kérdéspár ($K_{3.2.1}$ és $K_{3.2.2}$ kérdések) a 41. ábrához kapcsolódott:



41. 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 kérdéspáros ($K_{3.3.1}$ és $K_{3.3.2}$ kérdések) a 42. ábrán 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?



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

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

15. táblázat. *Kérdőív (Számítógépes gondolkodás vizsgálata különböző korosztályú tanulónál)*

Kérdőív részei	Tartalmi egységek	Kérdések száma
Első szakasz		
1. Blokk	Demográfiai adatok	1
2. Blokk	Informatika/tánc iránti szeretet	2
3. Blokk	Számítógépes gondolkodásuk szintjének mérése	1
4. Harmadik szakasz		
5. Blokk	Algoritmus megértése	1
6. Blokk	Bonyolultsági kérdések 5 elemű sorozatra	2
7. Blokk	Bonyolultsági kérdések x hosszúságú sorozatra	2

FÜGGELÉK

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) kicserélnék 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) kicserélnék 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, 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

Köszönetnyilvánítás

Nagy elismerésem és köszönetem illeti dr. Pluhár Zsuzsát, aki nélkül nem jöhetett volna létre és nem bővíthetett volna ki a nemzetközi Bebras kezdeményezés magyarországi megvalósítása a „Hódítsd meg a biteket” kezdeményezéssé.

A mellékletben található kérdéseket az általa létrehozott és támogatott http://e-hod.elte.hu/archiv/feladatok/hod2016_kishod.pdf dokumentum ihlette, és azzal a szenvedéllyel készültek, amit ő a projekthez hozzáadott.

B függelék: A buborék- és kiválasztó rendezések pszeudókódja

Buborékrendeztés 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);
```

Buborékrendeztés kódja a tánckoreográfia 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 tánckoreográfia 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]);
        }
    }
}
```

ÁBRAJEGYZÉK

1. ábra. Rekurzív függvény „eljátszva”	24
2. ábra. Buborékrendezés	25
3. ábra. Rendezési algoritmusok videói	25
4a. ábra. Új algoritmusok 2016-tól	27
4b. ábra. Új algoritmusok 2016-tól	28
5. ábra. Az AlgoRythmics új dimenziói	28
6. ábra. A rendszer architektúrájának diagramja	30
7. ábra. Használati eset diagramja	31
8. ábra. Tanulási lépések	32
9. ábra. Videótanulási lépés	32
10. ábra. Animációtanulási lépés	33
11. ábra. Levezényléstanulási lépés	34
12. ábra. Kódépítés-tanulási lépés	35
13. ábra. Életre kelt kód tanulási lépés	36
14. ábra. Kvíz tanulási lépés	37
15. ábra. Algoritmusok listája – adminisztrátori felület	38
16. ábra. Algoritmusok listája felhasználói felület	38
17. ábra. Tanfolyamok listája – adminisztrátori felület	39
18. ábra. Tanfolyamok listája – felhasználói felület	40
19. ábra. A két csoport előteszt- és utóteszteredményei	49
20. ábra. Buborék vs. Kiválasztó rendezés eredményei az utóteszten	50
21. ábra. Statikus vs. dinamikus ábrázolásmód	55
22. ábra. A csoportonkénti eredmények átlagainak ábrázolása	57
23. ábra. Előteszt és utóteszt eredményei csoportonként	64
24. ábra. A 3 programozási szinten levő diákok eredményei az utóteszt során, csoportonként	64
25. ábra. Az előteszt és utóteszt eredményei összesítve nemenként	65
26. ábra. Az előteszt és utóteszt eredményei nemenként csoportokra lebontva	65
27. ábra. Az 1–6., 7–8., 9–12. kérdések válaszai az 1 vagy 2 évet tanultak esetén	66
28. ábra. A diákok véleménye a leghatékonyabb tanulási módszerről	66
29. ábra. Lineáris keresés flamencotáncsal	70
30. ábra. Bináris keresés flamencotáncsal	70
31. ábra. A kísérlet első szakaszának eredményei korosztály szerint (mindkét iskola)	80
32. ábra. A kísérlet harmadik szakasz eredményei korosztály szerint (mindkét iskola)	81

33. ábra. A kísérlet harmadik szakaszának eredményei, az utolsó két kérdés kivételével, korosztály szerint (mindkét iskola)	82
34. ábra. A kísérlet első szakaszának eredményeinek feldolgozása iskolák szerint.	84
35. ábra. A kutatási kérdésekre adott válaszok feldolgozása nemenként	86
36. ábra. A Shell-rendeztés realisztikus ábrázolása	94
37. ábra. A Shell-rendeztés sematikus ábrázolása	94
38. ábra. A párját kereső fiú és a hételemű lány sorozat.	99
39. ábra. Lineáris keresés, videó.	100
40. ábra. Lineáris keresés, tízelemű sorozaton.	100
41. ábra. Lineáris keresés, best/worst case, 5 elemű számsorozaton	101
42. ábra. Lineáris keresés, best case/worst case, ismeretlen számú számsorozaton	101

TÁBLÁZATJEGYZÉK

1. táblázat. A Fisher-féle egzakt teszt eredménye	48
2. táblázat. Elő- és utóteszteredmények a két csoport szignifikanciakülönbségének mérésére	49
3. táblázat. A csoportonkénti eredmények átlagai és zárójelben a megfelelő szórásértékek.	57
4. táblázat. Résztvevők átlagai csoportokra bontva	63
5. táblázat. Az A csoport első és második szakaszbeli eredményei	73
6. táblázat. A kutatásban résztvevők száma korosztályonként, nemenként, iskolánként	77
7. táblázat. A kutatás kérdéseire adott válaszok átlaga és zárójelben a megfelelő szórásérték.	79
8. táblázat. Az első kérdésre adott helyes és helytelen válaszok száma osztályonként	80
9. táblázat. Adott iskolában tanult évek száma szakterületenként	83
10. táblázat. A kérdőív szerkezete (Sematikus versus emberi mozgást alkalmazó realisztikus algoritmusvizualizáció	92
11. táblázat. Kérdőív szerkezete (Sematikus és realisztikus algoritmusvizualizációk ismételt megtekintése, illetve ezek kombinálása)	95
12. táblázat. Kérdőív szerkezete (Az interaktivitás szerepe online algoritmusvizualizációs környezetekben)	96
13. táblázat. A kutatás lépései.	98
14. táblázat. Az A, B, C csoportok első szakaszbeli eredményei (mivel a Q1.2.6. kérdés megválaszolása viszonylag magas matematikai ismeretet igényelt, ezt kihagytuk elemzésünkben).	99
15. táblázat. Kérdőív (Számítógépes gondolkodás vizsgálata különböző korosztályú tanulóknál)	102

IRODALOMJEGYZÉK

- AHADI, A.–LISTER, R.–LAL, S.–LEINONEN, J.–HELLAS, A.
2017 Performance and Consistency in Learning to Program. *Proceedings of the Nineteenth Australasian Computing Education Conference*, ACM January 11–16.
- 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.
- XXX
2020 *AlgoRythmicsenvironment*. <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 The effect of student attributes on success in programming. *ACM SIGCSE Bulletin* 33. 3. 49–52.
- 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*. New York, NY, Springer, 551–580.
- CASTRO-ALONSO, J. C. –AYRES, P.–WONG, M.–PAAS, F.
2018 Learning symbols from permanent and transient visual presentations: Don't overplay 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 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*, ACM, June, 69–74.
- 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 Imagining inauthentic legitimate peripheral participation: an instructional design approach for motivating computing education. In: *Proceedings of the second international workshop on Computing education research*, ACM, September, 51–58.

- 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.–ELDMAN, 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 Selective hiding for improved algorithmic visualization. In: *Proceedings of the 2014 conference on Innovation & technology in computer science education*, ACM, June, 33–38.
- KÁTAI, 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.
- KÁTAI, 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 biofeedback. *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: Lowe, R.–Ploetzner, R. (eds.): *Learning from dynamic Visualizations: Innovations in research and application*, Berlin, Springer, 5–30.
- MANNILA, L.–DAGIENE, V.–DEMO, B.–GRGURINA, N.–MIROLO, C.–ROLANDSSON, L.–SETTLE, A.
2014 Computational thinking in K–9 education. In: *Proceedings of the working group reports of the 2014 on innovation & technology in computer science education conference*, ACM, 1–29.
- MAYER, R. E.
2005 Cognitive theory of multimedia learning. In: Mayer, R. E. (ed.): *The Cambridge handbook of multimedia learning*. Cambridge University Press, 31–48.
- 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 Commonsense computing (episode 5): algorithm efficiency and balloon testing. In: *Proceedings of the fifth international workshop on Computing education research workshop*, ACM, 51–62.
- 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 JHAVÉ – an environment to actively engage students in Web-based algorithm visualizations. *ACM SIGCSE Bulletin* 32. 1. 109–113.
- 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*. New York, NY, Springer, 605–622
- OSZTIÁN Pálma–Rozália–KÁTAI Zoltán–OSZTIÁN Erika
2021 AlgoRythmics online oktatási környezet: AlgoRythmics online learning environment. In: *Energetika-elektrotechnika – Számítástechnika és Oktatás Multi-konferencia*, 70–80.
- 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.

1997 *Neveléstörténet*. Bp., Nemzeti Tankvk., 185.

RAU, M. A.

2013 *Conceptual learning with multiple graphical representations: Intelligent tutoring systems support for sense-making and fluency-building processes* (Unpublished doctoral dissertation). Pittsburgh, PA, Carnegie Mellon University.

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 Beyond computer science: computational thinking across disciplines. In: *Proceedings of the 18th ACM conference on Innovation and technology in computer science education*, ACM, June, 311–312.
- 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 Commonsense computing: what students know before we teach (episode 1: sorting). In: *Proceedings of the second international workshop on Computing education research*, ACM, September, 29–40.
- SIMON, B.–KINNUNEN, P.–PORTER, L.–ZAZKIS, D.
2010 Experience report: CS1 for majors with media computation. In: *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education*, ACM, June, 214–218.
- STOUT, J.–TAMER, B.
2016 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*, ACM, February, 496–496).
- SWELLER, J.–VAN MERRIENBOER, 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 Impact of alternative introductory courses on programming concept understanding. In: *Proceedings of the first international workshop on Computing education research*, ACM, October, 25–35.
- 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 *International journal of human-computer studies* 57. 4.

WERTH, L. H.

1986 Predicting student performance in a beginning computer science class. *ACM* 18. 1. 138–143.

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.

FORRÁSJEGYZÉK

AlgoRythmicschannel

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

Bebras

2020 International Challenge on Informatics and 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>.

REZUMAT

Vizualizarea algoritmilor bazată pe efectul mișcării umane

În era digitală este acceptată ideea că Gândirea Computațională (GC) reprezintă o mentalitate fundamentală pentru toți elevii care învață. Termenul „gândire computațională” a fost introdus în 1980, dar abia în 2006, Jeannette Wing l-a adus în prim-planul comunității de educație în domeniul informaticii [Win12]. Wing descrie GC ca fiind o competență formală echivalentă cu cititul, scrisul și calculul matematic. Conceptul de GC poate fi definit ca un procesul cognitiv în care problemele sunt formulate în așa fel încât soluțiile lor pot fi descrise în pași computaționali, algoritmi [Aho12]. O abordare posibilă pentru a face dezvoltarea gândirii computaționale atractivă pentru diferite categorii de elevi (inclusiv elevi din învățământul primar și liceu, precum și cei care nu studiază informatică) este contextualizarea [GT06]. Mediul de învățare AlgoRhythms se bazează pe acest principiu. Deoarece muzica și dansul sunt relativ aproape de interesele majorității tinerilor, acest mediu AlgoRhythms vizualizează zece algoritmi fundamentali în informatică (căutarea și sortarea) prin coregrafii de dans profesioniste (dans popular, flamenco, balet). În plus față de combinarea motivantă a artei și științei, un alt avantaj al vizualizărilor este faptul că AlgoRhythms oferă o educație multisenzorială. Pe lângă aceste motivații poate că un lucru și mai important este faptul că algoritmii sunt ilustrați cu mișcări umane. Cele mai recente rezultate de cercetare cu privire la „efectele mișcării umane” subliniază că observarea mișcării umane (sau implicarea în activitate fizică) poate fi benefică din punct de vedere cognitiv [CAAWP18]. În concluzie, platforma AlgoRhythms este un mediu de învățare electronic interactiv și multisenzorial care promovează dezvoltarea abilităților de GC pentru diferite categorii de elevi. Acest scop este servit de studiul efectuat în noul mediu AlgoRhythms, prezentat în această carte.

ABSTRACT

Human-Oriented Algorithm Visualization

Nowadays it is commonly accepted that computational thinking (CT) is an essential mindset for all students of the digital era. Although the phrase computational thinking was introduced in 1980, it was brought to the forefront of the computer science (CS) education community only in 2006 by Jeannette Wing [Win12]. Wing describes CT as a formative skill on a par with reading, writing, and arithmetic, and she emphasizes that everyone, not just computer scientists, would be eager to learn and use. Since the concept of CT can be defined as the thought process involved in formulating problems so that “their solutions can be represented as computational steps and algorithms” [Aho12], a natural way to address the “CT for all” issue is computing education for all. A possible approach to make computing education attractive for different categories of learners (including K–12 learners and non-CS majors) is contextualization [GT06]. The AlgoRhythmics learning environment was designed along this approach. Since music and dance are relatively close to most young people, this environment visualizes ten basic computer algorithms (searching and sorting) by professional dance choreographies (folkdance, flamenco, ballet). Besides the motivating arts–science combination, another advantage of the AlgoRhythmics videos is that they activate multi-sensory learning. In addition, the algorithms are illustrated by human movement. Recent research results on the so-called human movement effect emphasize that observing human movements (or producing our own body movement) can be cognitively beneficial [CAAWP18]. It can be stated that AlgoRhythmics platform is a multisensory, interactive e-learning environment to promote the CT of different categories of learners. Therefore, I decided to implement my recent studies in this renewed AlgoRhythmics environment.

A SZERZŐRŐL

Osztian Erika Marosvásárhelyen született 1972-ben. Középiskolai éveit a Papiu Ilarian Elméleti Líceumban töltötte, ahol tanárainak köszönhetően a matematika és a fizika tantárgyak hamarosan a legkedveltebbekké váltak számára. Az érettségi után a Petru Maior Műszaki Egyetem Automatizálás és Számítógépek Műszaki Karán tanult, és sikeresen szerzett mérnöki diplomát.

Az oktatás iránti szeretete már fiatalon megmutatkozott. Címzetes tanárként kezdett tanítani az Elektromaros Szaklíceumban 2001 és 2016 között. Itt segített számos diáknak a programozás és informatika területén való előrehaladásban, és irányította a diákok tanulmányi és szociális fejlődését.

2016-ban csatlakozott a Sapientia Erdélyi Magyar Tudományegyetem matematika–informatika tanszékéhez, ahol tanársegédként kezdte tevékenységét. 2021-től adjunktusként a programozás, valamint az adatszerkezetek és az algoritmusok terén oktat és kutat.

KÖSZÖNETNYILVÁNÍTÁS

A könyv létrejöttében kiemelkedő szerepet játszottak dr. Kátai Zoltán, dr. Kása Zoltán és dr. Osztian Pálma Rozália. Ők valamennyien kulcsfontosságú szereplők voltak, akik meghatározóan hozzájárultak a könyv tartalmához és minőségéhez. Köszönöm.

Scientia Kiadó

400112 Kolozsvár (Cluj-Napoca)

Mátyás király (Matei Corvin) u. 4. sz.

Tel./fax: +40-364-401454

E-mail: scientia@kpi.sapientia.ro

www.scientiakiado.ro

Műszaki szerkesztés:

Metaforma Kft.

Borítóterv:

Tipotéka Kft.

Korrektúra:

Szenkovics Enikő

Tipográfia:

Könczey Elemér

Nyomdai munkálatok:

F&F INTERNATIONAL Kft.

Felelős vezető: Ambrus Enikő igazgató

A könyv a szerző doktori dolgozata, mely bepillantást nyújt a Sapiencia EMTE által kifejlesztett AlgoRhythmics tanulási környezetbe és ez ezzel kapcsolatos kutatásokba.

Az AlgoRhythmics tanulási környezet az informatika oktatását és a számítógépes gondolkodás fejlesztését szolgálja. Nemcsak a rendezési, hanem a keresési algoritmusokat is tánccoreográfiákkal ábrázolja, kiemelve a sematikus és realisztikus vizualizációk közötti különbségeket. Ezeket az ábrázolásokat az etnikumok néptáncai még inkább szemléletessé teszik.

A könyv részletesen ismerteti az AlgoRhythmics webes környezet fejlődését, beleértve az interaktivitási szint beállításának lehetőségét is. A könnyen érthető stílus, valamint az ábrák, a képek és a táblázatok gazdag tárháza segíti a témák megértését, így a könyv alap lehet az informatikaoktatásban és a további algoritmusvizualizációs kutatásokban egyaránt.

ISBN 978-606-975-085-8



9 786069 750858