

## AlgoRythmics online oktatási környezet

## AlgoRythmics online learning environment

OSZTIÁN Pálma-Rozália, Dr. KÁTAI Zoltán, Dr. OSZTIÁN Erika

Sapientia Erdélyi Magyar Tudományegyetem Marosvásárhelyi kar

1918. December 1 sugárút, Corunca 540485

telefon: 0265 208 181, honlap: <https://ms.sapientia.ro/hu>

email: [osztian.palma@ms.sapientia.ro](mailto:osztian.palma@ms.sapientia.ro), [katai\\_zoltan@ms.sapientia.ro](mailto:katai_zoltan@ms.sapientia.ro), [osztian@ms.sapientia.ro](mailto:osztian@ms.sapientia.ro)

### Abstract

*Nowadays, the use of online educational environments is very widespread, the importance of which has been further highlighted by the viral situation that has arisen in the recent period. We are increasingly finding that their use is essential, as we often have to move into the online world for reasons beyond our control.*

*In 2011, an educational environment was created that presents different sorting and searching algorithms in a special, unusual way. Since then, related teaching materials have undergone significant development and the method has enjoyed great popularity. This year, on the occasion of the 10th anniversary of the environment, we would like to announce a new, complex and spectacular educational environment, which is also an improved version of the previous one. The environment uses 6 learning steps to guide users from dance to code using a variety of algorithms and courses. It also provides opportunities for both teaching (teachers) and learning (learners).*

**Keywords:** algorithms, E-learning, interactivity, learning steps

### Kivonat

*Napjainkban igen elterjedt az online oktatási környezetek használata, melyek fontosságára az elmúlt időszak során felmerülő virushelyzet még inkább rávilágított. Egyre inkább azt tapasztaljuk, hogy ezek használata elengedhetetlen, hiszen sokszor tőlünk független okokból át kell költöznünk az online világba.*

*2011-ben létrejött egy olyan oktatási környezet, amely különleges, a szokásostól eltérő módon mutat be informatikai algoritmusokat. Azóta, az ehhez kapcsolódó tananyagok jelentős fejlődésen mentek keresztül és a módszer nagy népszerűségnek örvend. Idén, a környezet 10 éves évfordulója alkalmából egy új, komplex és látványos oktatási környezetet szeretnénk közzétenni, amely egyben az előző környezet egy továbbfejlesztett változata. A környezet 6 tanulási lépés segítségével vezeti el a felhasználókat a táncról egészen a kódig különféle algoritmusok és tanfolyamok segítségével. Emellett lehetőséget biztosít úgy a tanításra (tanárok), mint a tanulásra (tanulók) egyaránt.*

**Kulcsszavak:** algoritmusok, E-learning, interaktivitás, tanulási lépések

## BEVEZETÉS

Ha oktatásról, informatikáról és algoritmusokról beszélünk elengedhetetlen megemlíteni az online oktatási környezetek jelenlétét. Jelen napjaink oktatási stratégiája nagymértékben támogatja ezen rendszerek használatát. Sokszor társítható egy-egy hagyományos, osztályteremben bemutatott algoritmus magyarázatához egy kifejező animáció vagy videó, amely még inkább segíti a tanulókat az algoritmus megértésében és elsajátításában.

Az AlgoRythmics online oktatási környezet első kulcsfontosságú elemei, a tánckoreográfia videók egy része már 2011-ben elérhető volt. A videók a szokásostól eltérő módon, erdélyi magyar etnikumok segítségével szemléltettek különböző rendező és kereső algoritmusokat. Az idők során

bebizonyosodott, hogy ezek a tánckoreográfia videók nagy népszerűségnek örvendenek éppen ezért továbbfejlesztésre volt szükség. Ekkor jött létre egy olyan oktatási környezet, amely a tánckoreográfia videók mellett animációk segítségével is vizualizálták az algoritmusokat. Később mindezt egy új szintre emeltük, és társítottunk további 3 tanulási lépést a már meglévő vizualizációkhoz, és a tánckoreográfia videók száma is 10-re növekedett. Az így kialakított online oktatási környezet lehetőséget biztosított úgy az algoritmusok elsajátítására, mint különböző kutatási témák kivitelezésére is. Az elmúlt néhány év során számos kutatás látott napvilágot, melyeknek köszönhetően új tapasztalatokat nyerhettünk úgy az oktatási stratégia, mint a környezet nyújtotta lehetőségek kiaknázása szempontjából. Célunk az volt, hogy 2021-ben a környezet egy olyan új változatát mutassuk be a nagyközönség előtt, amely megnyitja a kaput más intézmények oktatói felé és lehetőséget biztosít személyre szabott tanfolyamok létrehozására.

Az előző kutatási eredményeket és eddigi tapasztalatainkat figyelembe véve, reményeink szerint az új környezet segít majd a tanulóknak megérteni és megszeretni az alapvető informatikai algoritmusokat, továbbá segít a más egyetemekkel és iskolákkal való kapcsolatok kialakításában, akik előtt megnyitjuk az környezet oktatói oldalát.

## SZAKIRODALMI TANULMÁNY

Az elmúlt időszak során a környezet folyamatos fejlesztésen ment keresztül, valamint számos kutatásnak biztosított lehetőséget. Annak érdekében, hogy folyamatosan törekedjünk a tanfolyamok és az oldal felhasználóbarát, valamint tartalmi hozadékára elengedhetetlen volt megvizsgálni az ehhez kapcsolódó szakirodalmi kutatásokat.

Mivel a környezet elsődleges célja, hogy a videók mellett animációk segítségével szemléltessen különböző rendező- és kereső algoritmusokat, elsősorban ezek fontosságára szerettünk volna hangsúlyt fektetni. A környezet egy további kulcsfontosságú jellemzőjének számít az interaktivitás, melynek fontosságát számos kutató hangsúlyozta ki. Az interaktív algoritmus vizualizációval (AV), és azok fejlesztésével régóta foglalkoznak a kutatók. Naps és munkatársai [1] például már 2000-ben megfogalmaztak egy ezzel kapcsolatos következtetést, miszerint az interaktív vizualizációk segítenek a tanulóknak abban, hogy elsajátítsák a számítógépes gondolkodás alapfogalmait. Ugyancsak ez a kutatócsoport fogalmazott meg hat kulcsfontosságú alapelvet, amelyek az aktív tanulásra vonatkozó szinteket határozzák meg: **nincs megtekintés**, **megtekintés** (a hallgatók passzívan figyelik meg az AV-t), **válaszadás** (a hallgatók figyelik az AV-t és bizonyos kulcsmomentumokban válaszolnak a vizualizáció során felmerülő kérdésekre), **változtatás** (a hallgatók megváltoztatják az AV-t például az algoritmus bemeneti adatainak meghatározásával), **felépítés** (a hallgatók építik fel az AV-t) és **bemutató** (a hallgatók bemutatják az AV-t társaiknak).

Továbbá, egy Grissom, McNally és Naps [2] által kidolgozott kutatás összehasonlította három csoport eredményét, akik az aktív tanulás különböző szintjén bemutatott vizualizációkkal tanultak: **vizualizáció nélküli** (nem tartalmazott vizualizációt), **megtekintés** (a vizualizáció egyszerű megtekintése), **válaszadás** (felhasználói interakciók a vizualizáció során). A kutatók azt a következtetést vonták le, hogy minél magasabb szintű felhasználói bevonást alkalmaztak, annál jobb tanulási eredményeket értek el. Mindezek megerősítettek abban, hogy az új oktatási környezet elengedhetetlen elemei közé kell tartozzon az interaktivitás. Ennek fontosságáról számos kutatásunk során sikerült megbizonyosodni ([3], [4], [5]).

Továbbá, mivel a környezet az algoritmusok típusának függvényében azok két különböző csoportját (iteratív és rekurzív) is szemlélteti, fontos volt megvizsgálni az iteratív és rekurzív algoritmusok vizualizációjára vonatkozó különbségeket is.

A **rekurzió** kapcsán számos kutatás látott napvilágot [6], melyek elsősorban arra összpontosítottak, hogy mit is jelent, mit foglal magába a rekurzió és hogyan lehetne ezt hatékonyan tanítani. Több kutató is kihangsúlyozza, hogy a rekurzió fogalma homályos és következtetlenül a program egyes szintaktikai tulajdonságaiból, a hozzá kapcsolódó szemantikából és a programok végrehajtási modelljeiből kölcsönzött funkciókból épül fel [7], míg mások olyan folyamatként jellemzik, mely képes újabb eredményeket kiváltani önmagától, melyeknek folyamatosan átadja a kontrollt, míg végül visszatér a befejezett feladatoktól [8].

A különböző megfogalmazások és részben eltérő meghatározások ellenére egy dologban azonban sokan egyetérthetünk: a rekurzió nem egy egyszerű folyamat, és korántsem könnyű ennek tanítása,

főként kezdők számára. Azt is nehéz meghatározni, és már régóta vitatott témának számít a kutatók körében, hogy mikor lenne a leginkább optimálisabb és hatásos ennek tanítása ([9], [10], [11]). Ha elfogadjuk azt, hogy az algoritmus-vizualizáció jelentősen javíthat az eredményeken és elősegítheti a megértést, akkor természetesen ez a rekurzió esetén sem idegen technika. Mindezt az is alátámasztja, hogy a rekurzióhoz kapcsolódó tananyag gyakran játékok, matematikai feladatok kíséretében kerül terítékre, mint például Fibonacci számsorozat, Hanoi tornyai, Pascal háromszög ([12], [13], [14]). Az érdekes és olykor már ismerősebb fogalmak elősegíthetik úgy a tanulók kíváncsiságának felkeltését, mint az algoritmusban, rekurzióban való elmélyülést is, és ennek az élménynek és könnyed „tálatásnak” köszönhetően talán azt is elfelejtik a diákok, hogy tanulnak. Ezáltal a rekurzió elsajátítása pedig egy sokkal természetesebb folyamattá válik. Ennek egy példájával él Papert [15], aki úgynevezett trükknek nevezi a rekurziót, melyet a tanárok „meseszerűen” kellene tálataljanak, hiszen a rekurzióban fellelhető végtelen, minden gyerek fantáziáját megéri (,„... ha van két kívánságod, mi lenne a második? Még két kívánság.”).

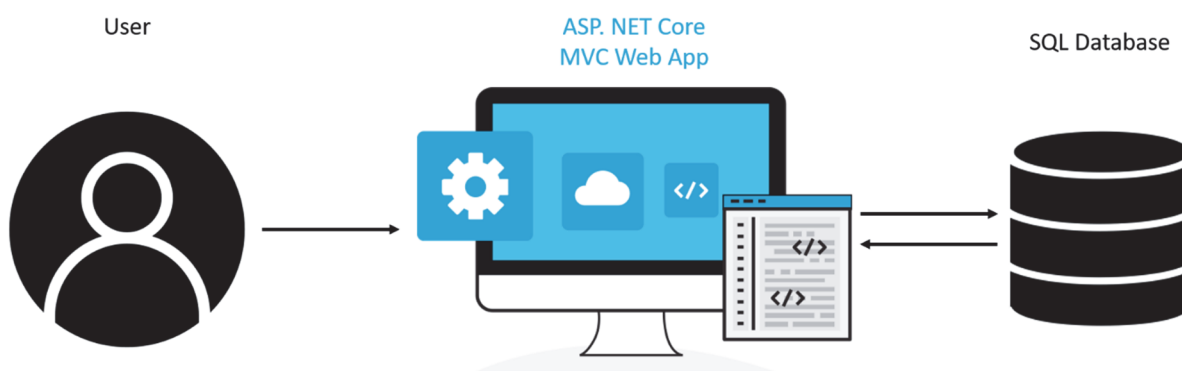
A rekurzió megértésének elősegítése érdekében gyakran hozzák párhuzamba ezt az **iteratív** algoritmusok fogalmával. Piaget [16] például a dominók sorozatának esésével szemlélteti ezt a párhuzamot, mely egyaránt tekinthető rekurzív (minden eldőlt dominó az előző eldőlt dominó következménye), de iteratív (eldől az első dominó, majd ezt követően a második és így tovább...) eljárásnak is. Mindez abban rejlik, hogyan is szemlélteti ezt a tanár, illetve, hogy mit biztosít az algoritmus-vizualizációs környezet.

Stern és Naish [17] osztályozása alapján a rekurzív rendező algoritmusok jelentős részét képezik a rekurzív eljárásoknak és ezek egyik leghasznosabb és legjellegzetesebb megjelenítési formája közé tartozik az animáció. Ennek segítségével könnyen és látványosan szemléltethető, valamint nyomon követhető a rekurzióban fellelhető fa-struktúra, de egyéb adatszerkezetek is [18].

Mindezeket figyelembe véve, arra összpontosítottunk, hogy az új online oktatási környezet a korábban említett **interaktivitás** mellett, a **rekurzív** és **iteratív** algoritmusok megjelenítésére is biztosítson lehetőséget, úgy, hogy az kifejező, érthető legyen a felhasználók számára.

## A RENDSZER LEÍRÁSA

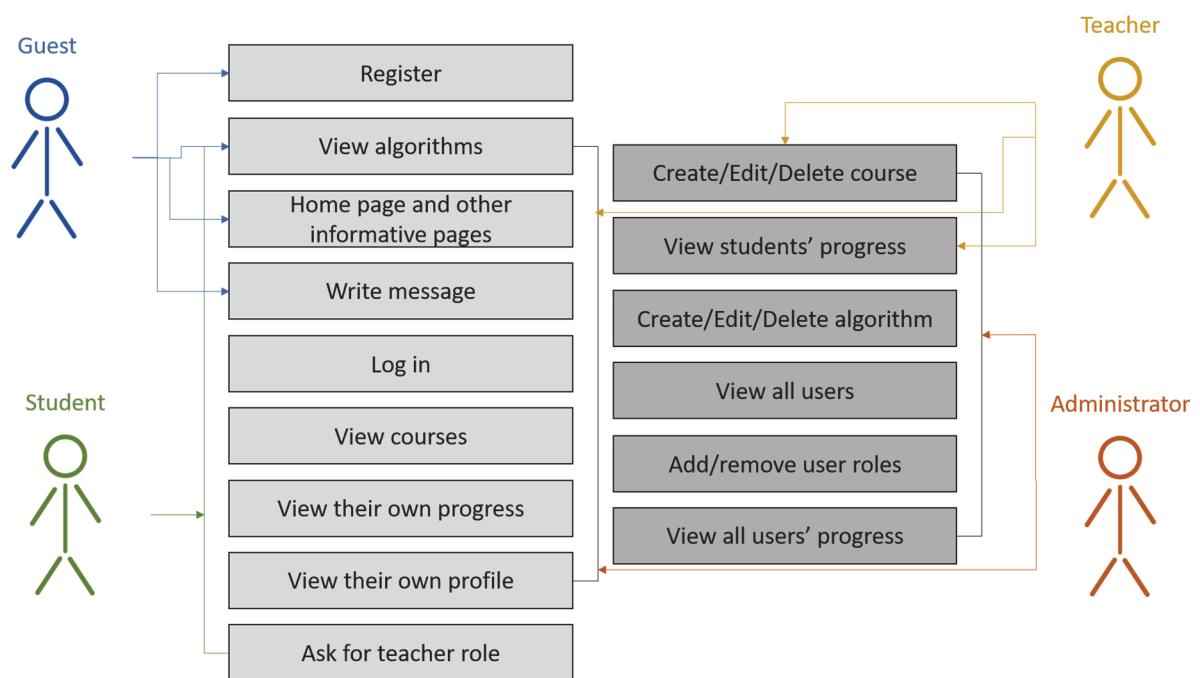
Amint azt a rendszer architektúrája is szemlélteti (1. ábra), a felhasználók az interneten keresztül érik el a webalkalmazást. Az alkalmazás egy SQL adatbázissal rendelkezik, amely felhasználásával szolgáltatja az adatokat a kliensnek.



1. ábra. A rendszer architektúrája

Az alkalmazást négy különböző felhasználó használhatja: adminisztrátor, tanár, diák, illetve vendég. Amint az a rendszer használati eset diagramján is látható a vendég felhasználó megtekintheti az alkalmazáshoz kapcsolódó informatív jellegű oldalakat (pl. Főoldal, Elérhetőségek, Közösségi oldalak),

az algoritmusok listáját és az algoritmusokhoz tartozó alap tanfolyamokat, valamint lehetősége van bejelentkezni. A bejelentkezett felhasználó közvetlen a bejelentkezés után tanulói jogot kap, amely azt jelenti, hogy lehetősége van az elérhető tanfolyamok elvégzésére, megtekintheti saját, a tanfolyamokon való előrehaladását, valamint a felhasználójához tartozó személyes információkat. Emellett, rendelkezik minden joggal, amely a vendég felhasználót is megilleti. A következő felhasználói kör a tanároké. Minden bejelentkezett felhasználó, aki tanárként szeretné használni a webalkalmazást jeleznie kell szándékát az adminisztrátornak. Ezt megteheti a webalkalmazás felületén, vagy az elérhetőségek felkeresésének segítségével. A tanár rendelkezik minden korábban felsorolt joggal, emellett azonban lehetősége van saját tanfolyamok létrehozására. Minden tanfolyam létrehozása esetén választhat a saját, rejtett tanfolyam, illetve publikus tanfolyamok között. A tanár továbbá nyomon követheti saját tanfolyamához tartozó tanulói előrehaladást. Az adminisztrátornak minden korábban említett lehetőség mellett még jogában áll kezelni az algoritmusok listáját is. Emellett, a felhasználók jogait is módosíthatja, tanári jogot adhat, illetve vehet el a felhasználóktól. További feladatkörei közé tartozik az is, hogy minden nyilvánosnak szánt tanfolyamot jóváhagyjon.



2. ábra. A rendszer használati eset diagramja

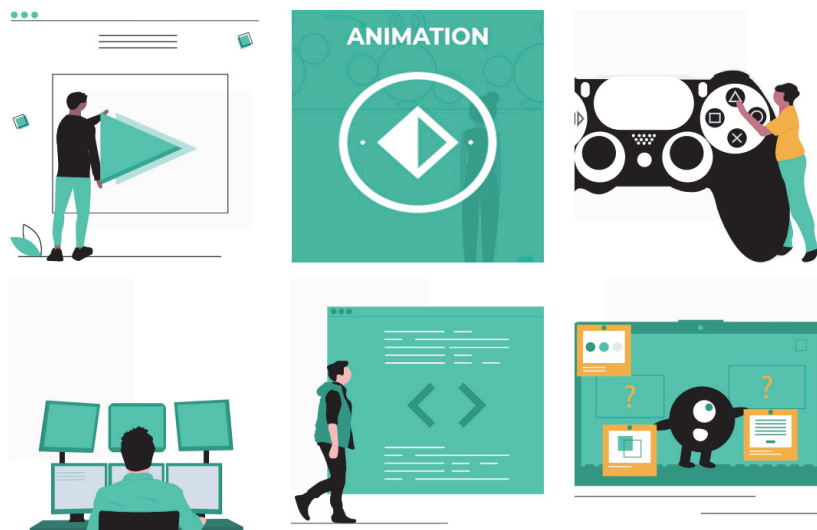
## A RENDSZER ALAPVETŐ ELEMEI

Ahhoz, hogy a webalkalmazás kreatívan és változatosan mutassa be a felhasználók számára az algoritmusokat és az azokhoz tartozó tanfolyamokat fontos volt meghatározni és alaposan kidolgozni az alkalmazás alapvető egységeit: a tanulási lépéseket, az algoritmusokat és a tanfolyamokat.

### 4.1. Tanulási lépések

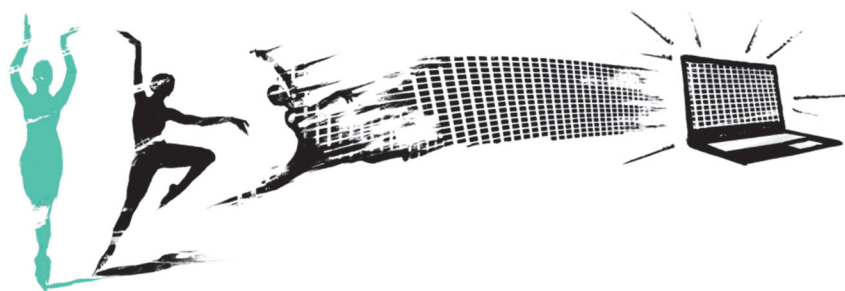
Az alkalmazás jelenleg 6 alapvető tanulási lépés (3. ábra) segítségével segít elsajátítani egy adott algoritmust. A tanulási lépések alap sorrendje egy úgynevezett ajánlott sorrend, amely nehézségi szintek szerint vonultatja fel a lépéseket (két megértést elősegítő lépés, két kódolást elősegítő lépés, egy összefoglaló jellegű lépés).

## You can choose from various learning steps



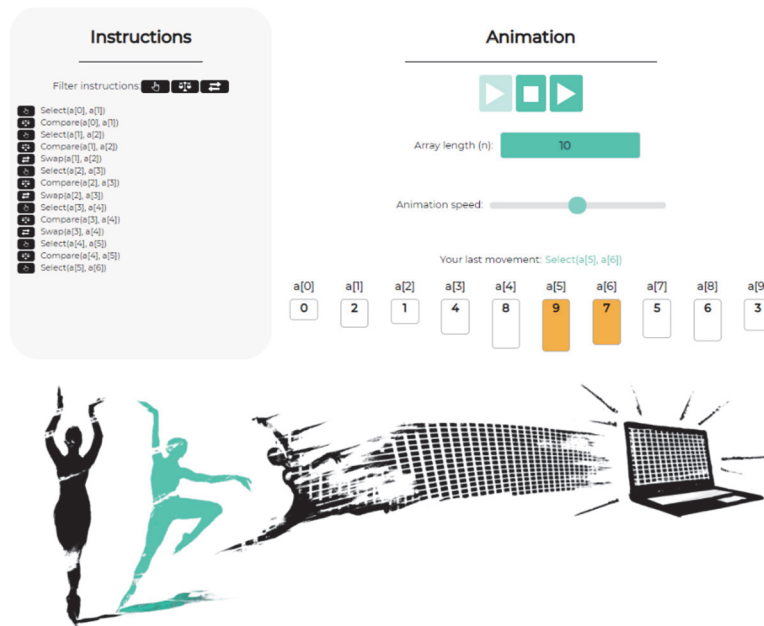
3. ábra. Tanulási lépések - főoldal

A videó: egy megértést elősegítő tanulási lépés, amely tartalmazza az algoritmust bemutató tánckoreográfiát (4. ábra). Alapvetően automatikus videóról beszélhetünk, viszont külön létrehozott tanfolyamok esetén interaktív videót is társíthatunk az adott tanmenethez. Ebben az esetben a videó bizonyos, előre meghatározott kulcsmomentumokban megáll és felhasználói interakció szükséges.



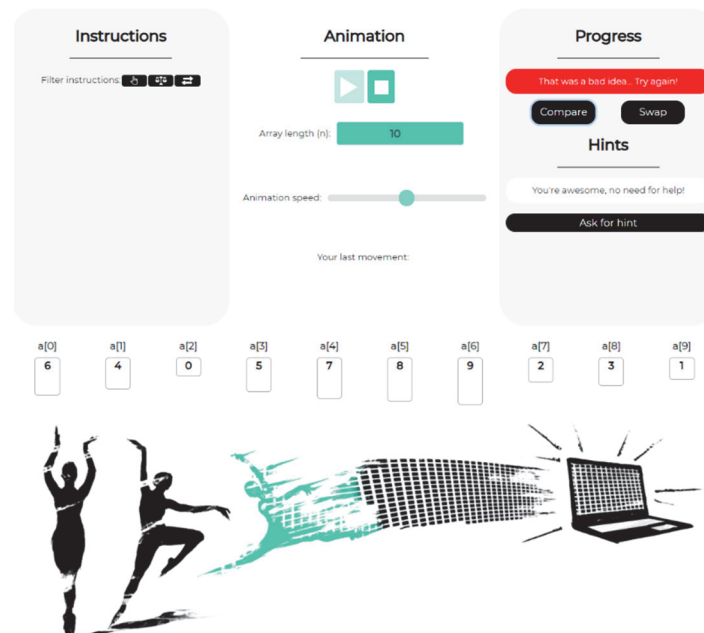
4. ábra. Videó tanulási lépés

Az animáció: az algoritmusvizualizáció egy absztraktabb változata (5. ábra), amely hasonlóképpen az algoritmus megértését elősegítő tanulási lépés. A videóhoz hasonlóan az animáció is lehet automatikus vagy interaktív, valamint előre meghatározható a bemeneti számsorozat jellege is (legjobb-, legrosszabb eset, véletlenszerű, konkrét számsorozat). Emellett, lehetőség van úgynevezett látható (a számsorozat elemeinek értéke látható) valamint láthatatlan (a számsorozat értékei rejtve maradnak) megjelenítést választani.



5. ábra. Animáció tanulási lépés

A levezénylés: az animáció egy továbbfejlesztett változata, amely segítségével a felhasználónak lehetősége van levezényelni az algoritmushoz tartozó animációt elejétől egészen a végéig (6. ábra). Ez azt foglalja magába, hogy a felhasználó saját belátása szerint válassza ki a soron következő összehasonlítandó/ cserélendő elemeket. A bemenet típusa, valamint a megjelenítési forma szerint is, az animációhoz hasonlóan ez a tanulási lépés is lehetőséget biztosít arra, hogy különféle módokban működjön.



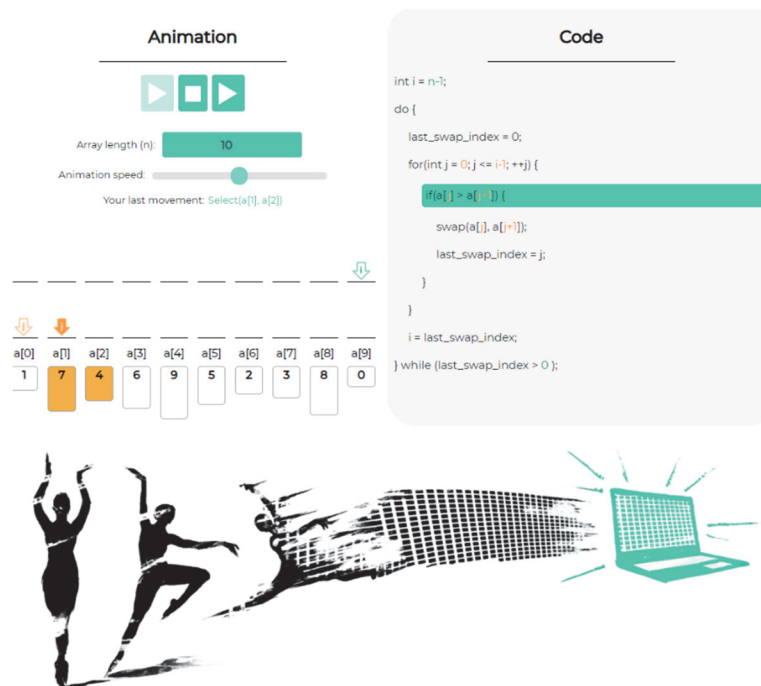
6. ábra. Levezénylés tanulási lépés

A kódépítés: az első olyan tanulási lépés, amely során láthatóvá válik az algoritmushoz tartozó kód (7. ábra). Az animáció segítségével a felhasználónak ezúttal a megjelenített kódrészletet kell kiegészítenie (ciklusátárok, feltételek). Ez a tanulási lépés már a kódolásra is összpontosít.



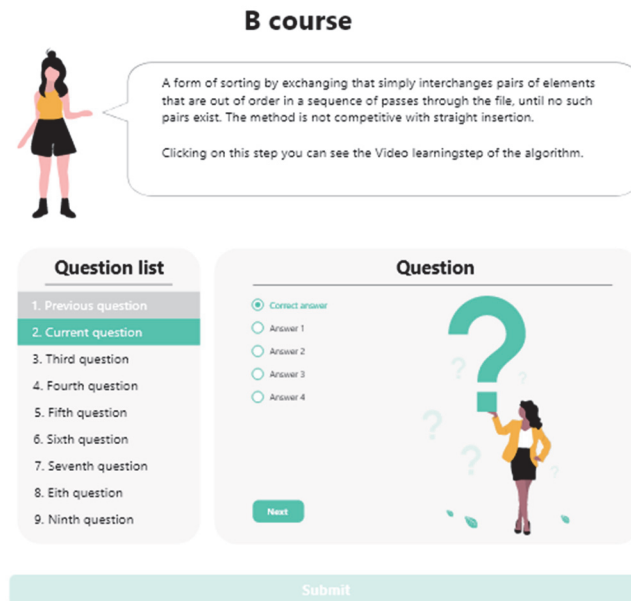
7. ábra. Kódépités tanulási lépés

Az életre kelt kód: az előző 4 tanulási lépés úgynevezett összefoglalójaként is tekinthető (8. ábra). Ebben a tanulási lépésben az animációt az előző lépések során kiegészített kód mozgatja. A kód különböző részei az animáció adott lépéseivel összhangban hangsúlyozódik ki.



8. ábra. Életre kelt kód tanulási lépés

A quiz: az elsajátított tudás felmérésére szolgáló tanulási lépés. Célja, hogy a korábban elvégzett tanulási lépések során felhalmozódott tudást felmérje (9. ábra).



9. ábra. Kvíz kérdések

## 4.2. Algoritmusok

A korábban felsorolt tanulási lépések összessége határozza meg egy algoritmus vizualizációját. Az oldalon megtalálhatóak egyaránt rendező (*buborék-, minimum kereső-, shell-, beszűrő-, összefésülő-, gyors rendezés*) és kereső (*lineáris-, bináris keresés*) algoritmusok, valamint az algoritmusok típusa függvényében is elkülöníthetünk úgy iteratív, mint rekurzív algoritmusokat.

**Algorithms** [Add new algorithm](#)

ICON	NAME	TYPE	STATUS	ACTIONS
	Bubble sort	Sorting	Unpublish	<a href="#">Edit</a> <a href="#">Delete</a>
	Insertion sort	Sorting	Unpublish	<a href="#">Edit</a> <a href="#">Delete</a>
	Selection sort	Sorting	Unpublish	<a href="#">Edit</a> <a href="#">Delete</a>
	Shell sort	Sorting	Unpublish	<a href="#">Edit</a> <a href="#">Delete</a>
	Linear search	Searching	Unpublish	<a href="#">Edit</a> <a href="#">Delete</a>

10. ábra. Algoritmusok listája - Adminisztrátori felület

Az algoritmusok listája a rendező, kereső és visszalépéses kereső algoritmusok csoportjai szerint van felosztva (11. ábra). Egy-egy algoritmus vizualizációjára lépve a felhasználónak lehetősége van megtekinteni az algoritmushoz tartozó alap tanfolyamot. Ez egy-egy tanulási lépést foglal magába, melyek elvégzéséhez nem kötelező, de ajánlott a lépések betartása. Az algoritmus során vétett hibák úgynevezett ismeretlen felhasználóként kerülnek elmentésre 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 part are inserted into the sorted part one by one.

Read more

11. ábra. Algoritmusok listája

### 4.3. Tanfolyamok

Amennyiben saját, specifikus tanfolyamokat szeretnénk létrehozni segítségül szolgál a tanfolyamok létrehozó felület, mely a tanárok és az adminisztrátor által elérhető (12. ábra).

Courses Add new course

My courses All courses Default courses

Bubble sort base course  
Bubble desc

Insertion sort base course  
Insertion description

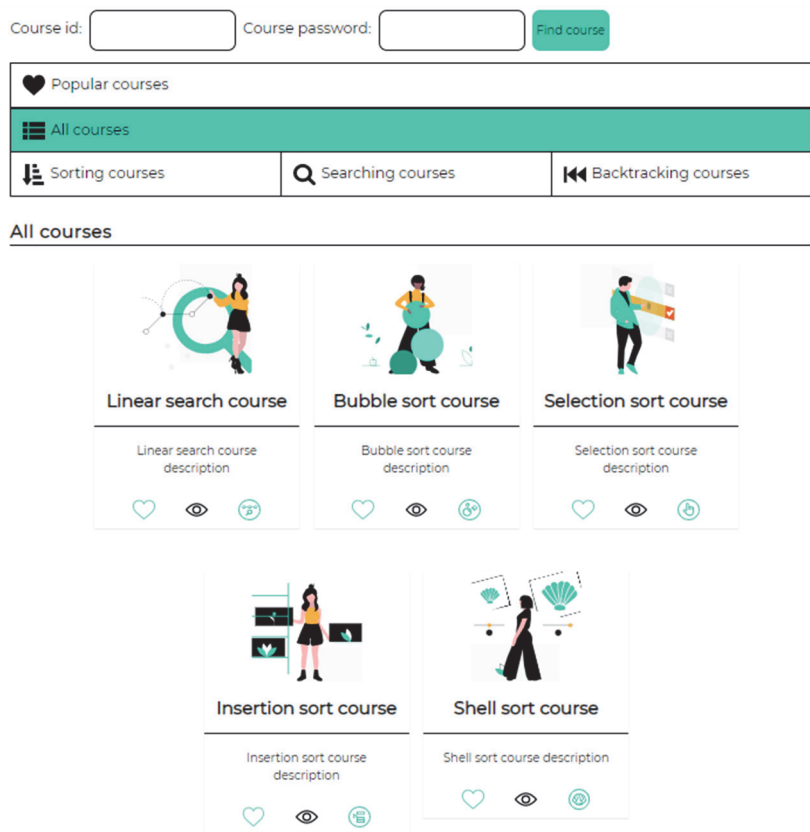
Selection sort base course  
The selection sort algorithm s

Shell sort base course  
ShellSort is mainly a variatio

Linear search base course  
In computer science, a linear

12. ábra. Tanfolyamok listája - Tanári felület

Egy adott tanfolyam az algoritmusok alapvető tanmeneteihez hasonló, viszont ebben az esetben a tanár határozhatja meg szabadon a tanulási lépések számát, milyenségét és azok sorrendjét. Ezáltal, arra is lehetőség van, hogy egy tanfolyam során csupán az animáció különböző formáit, vagy éppenséggel a kódírást használja az oktató.



13. ábra. Tanfolyamok listája

## ÖSSZEFOGLALÓ

Az elmúlt évek során számos kutatást sikerült megvalósítani a jelenleg működő **AlgoRhythmic** környezet segítségével. Ezekből és a diákokkal való közös munka során rengeteget tanulhattunk, sokat fejlődhattunk és fény derült a környezet számos hiányosságára, hibájára, de természetesen a benne rejlő potenciálra is.

Az elmúlt időszakban ennek továbbfejlesztése kiváltképp fontos volt számunkra, és a módosítások során igyekeztünk a lehető legtöbb újdonságot beépíteni a webalkalmazásba. Bár továbbfejlesztésre mindig van igény és lehetőség, és ez alatt az időszak alatt is számos újdonság lehetősége fogalmazódott meg bennünk, reméljük, hogy az új környezet hasznos lesz úgy a tanulók, mint oktatók számára és előszeretettel fogják használni. Továbbá, azt is reméljük, hogy az így kialakított kapcsolatok segítségünkre lesznek és az oldalt használó oktatók további hasznos tanácsokkal tudnak ellátni majd minket, ami a környezetet illeti.

## IRODALMI HIVATKOZÁSOK

- [1] Thomas L Naps, James R Eagan, and Laura L Norton. Jhavi-an environ-ment to actively engage students in web-based algorithm visualizations, 2000.
- [2] Grissom, S., McNally, M. F., & Naps, T. (2003, June). Algorithm visualization in CS education: comparing levels of student engagement. In Proceedings of the 2003 ACM symposium on Software visualization (pp. 87-94).
- [3] Eszter Jáhel Nagy, Palma Rozália Osztián, Cristian Cosma, Zoltán Kátai, and Erika Osztián. Looking for the optimal interactivity level in thealgorithms learning...EdMedia + Innovate Learning, 2019:106–114,2019.
- [4] Palma Rozalia Osztian, Zoltan Katai, and Erika Osztian. Algorithm visu-alization environments: Degree of interactivity as an influence on student-learning. volume 2020-October. Institute of Electrical and ElectronicsEngineers Inc., 10 2020.

- [5] Osztián Pálma Rozália, Osztián Erika, and Kátai Zoltán. Algoritmus-vizualizációs környezetek: Az interaktivitás tanulási eredményekre valóhatása, 2020.
- [6] Rinderknecht, C. (2014). A survey on teaching and learning recursive programming. *Informatics in Education-An International Journal*, 13(1), 87-120.
- [7] Give'on, Y. S. (1990). Is recursion well defined??. *Computers & Education*, 14(1), 35-41.
- [8] Kahney, H. (1983). What do novice programmers know about recursion. In: *Proceedings of the Conference on Human Factors in Computing Systems*. ACM SIGCHI, ACM Press, Boston, Massachusetts, USA, 235–239
- [9] Olson, A.T. (1987). The curricular implications of recursion. In: *Proceedings of the Third International Conference for LOGO and Mathematics Education*. Concordia University, Montréal, Canada
- [10] Barfurth, M., Retschitzki, J. (1987). The pedagogical relevance of children working with recursion. In: *Proceedings of the Third Conference “Logo and Mathematical Education”*. Montréal, Canada, 3, 164–172.
- [11] Greer, J.E. (1989). A comparison of instructional treatments for introducing recursion. *Computer Science Education*, 1(2), 111–128.
- [12] Rubio-Sánchez, M., Hernán-Losada, I. (2007). Exploring recursion with Fibonacci numbers. In: *Proceedings of the Twelfth Annual Conference on Innovation and Technology in Computer Science Education*. ACM SIGCSE-SIGCUE, ACM Press, Dundee, Scotland, United Kingdom, 359–359.
- [13] Rubio-Sánchez, M., Pająk, B. (2006). Fibonacci numbers using mutual recursion. In: Salakoski, T., Mäntylä, Mikko, L. (Eds.), *Proceedings of the Fifth Annual Finnish/Baltic Sea Conference on Computer Science Education*, 41. TUCS General Publications, Finland, 174–177.
- [14] Rubio-Sánchez, M., Urquiza-Fuentes, J., Pareja-Flores, C. (2008). A gentle introduction to mutual recursion. In: *Proceedings of the Thirteenth Annual Conference on Innovation and Technology in Computer Science Education*. ACM SIGCSE-SIGCUE, ACM Press, Madrid, Spain, 235–239.
- [15] Papert, S. (1980). *Mindstorms: Children, Computers and Powerful Ideas*. First edn, Basic Books.
- [16] Piaget, J., Stratz, C. (1974). *La chute récurrentielle de dominos alignés*. In: *Réussir et comprendre*. Presses Universitaires de France, 21–33.
- [17] Stern, L., Naish, L. (2002a). *Animating recursive algorithms*. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, 4(2).
- [18] Velázquez-Iturbide, J.A., Pérez-Carrasco, A. (2010). *InfoVis interaction techniques in animation of recursive programs*. *Algorithms*, 3(1), 76–91.