

Problémák, kérdések az algoritmus vizualizációval kapcsolatban

Bende Imre

beiraai@inf.elte.hu

ELTE IK

Absztrakt. Az algoritmus vizualizációs eszközöket az oktatók nem, vagy csak ritkán használják az óráikon. A cikkben ennek miértjét szeretném megvizsgálni, és megtalálni a felmerülő kérdéseket, problémákat az eszközzel, és annak használatával kapcsolatban. Ezeket nagyobb pontokba szedve teszem meg. Befejezésül pedig egyfajta választ, megoldást is szeretnék adni, aminek eredménye lehet egy olyan algoritmus vizualizációs eszköz, weboldal gondolata, mely igyekszik minden felsorolt kérdést megoldani, így az oktatók már könnyedén építhetik be tanóráikba a segédanyag használatát.

Kulcsszavak: algoritmus vizualizáció, programozásoktatás, tanuláshatékonyosság, motiváció

1. Bevezetés

„Az algoritmus-vizualizáció (AV) a szoftver-vizualizáció alosztályaként számítógépes algoritmusok magas szintű működésének illusztrálásával foglalkozik, általában abból a célból, hogy a programozást tanulók jobban megértsék az algoritmus eljárásainak működését.” [10] Manapság a technológia adott arra, hogy a kicsit is hozzáértő fejlesztők, tanárok, diákok könnyedén létre tudjanak hozni algoritmus animációkat, algoritmus vizualizációkat (rövidítve: AV). Sőt, jó pár weboldal, szoftver (mind magyar, mind angol nyelven) is létezik már, ahol rengeteg ilyet találhatunk. Némelyik már nagyobb gyűjteménnyel is rendelkezik, amit bárki bővíthet a saját munkájával, így növelve annak méretét. Azonban az oktatók mégsem, vagy csak ritkán használják az óráikon ezeket az eszközöket. A cikkben ennek miértjét szeretném megvizsgálni, és megtalálni a felmerülő kérdéseket, problémákat az eszközzel, és annak használatával kapcsolatban. Ezeket nagyobb pontokba szedve teszem meg. Befejezésül pedig egyfajta választ, megoldást is szeretnék adni, aminek eredménye lehet egy olyan algoritmus vizualizációs eszköz, weboldal gondolata, mely igyekszik minden felsorolt kérdést megoldani, így az oktatók már könnyedén építhetik be tanóráikba a segédanyag használatát.

2. Felmerülő problémák, kérdések

C. D. Hundhausen [2] cikkében főbb szempontokat fogalmazott meg, amik miatt nem használják a tanárok az AV technológiát (tanárok válaszai alapján; a pontokat Végh Ladislav [11] fordításában sorolom fel):

- „Úgy érzik, nincs idejük utánanézni és megtanulni a használatukat.” [11]
- „Úgy érzik, hogy nincs idő foglalkozni velük az órán, mivel ez más tanórai tevékenységtől venné el az időt.” [11]
- „Úgy érzik, hogy animáció előkészítése a tanórákra túl sok erőfeszítésbe és időbe telik.” [11]
- „Úgy érzik, hogy a használatuk nem hatékony az oktatásban.” [11]

Amikor az irodalomban felmerül a kérdés, hogy miért is nem használják a tanárok az eszközt az óráikon, akkor ennek a 2003-as felmérésnek az eredményére hivatkoznak a cikkek írói (például: [4], [8], [11]). Ezeket a pontokat próbáltam szétszedni, ismertetni, illetve újabbakat is hozzáfűzni, amik csak adott környezetben lehetnek problémák.

2.1. Nem ismerik

Az algoritmus vizualizációnak jelenleg még nincsen túl nagy történelme, de az utóbbi tíz évben hatalmas fejlődésen ment keresztül és jelentősebb eszközök születtek, amelyek ezt a módszert használják. Azonban az első jelentősebb ok, amiért egyáltalán nem használják a tanárok, hogy nem is hallottak erről az oktatási segédanyagról.

2.2. Technikai előfeltételek

Eszköztől függően kellhet egy erősebb hardver a használatához. Valamint szoftvertámogatásra is szükség lehet, amiknek feltelepítése időigényes lehet, akár a tanárnak, akár az iskola rendszergazdájának. Bizonyos esetekben ez a plusz munka is eltántoríthatja a tanárokat az AV használatától, bár ez a legtöbb esetben ma már nem lehet probléma. Sajnos, bár ritka esetben, de előfordulhat olyan is, hogy nem minden diáknak van saját gépe az informatikateremben, így ki sem tudnák próbálni önállóan az algoritmusok használatát a vizualizáció segítségével.

2.3. Eszközök megismerése

Ahhoz, hogy egy új eszköz segítségével taníthassunk, azt magunknak is megfelelően meg kell ismerünk. Ehhez azonban szükséges az, hogy a szoftverhez legyen egy jó leírás, dokumentáció, illetve a szoftver használata egyértelmű, tiszta legyen. Ez azonban sok esetben nem valósul meg.

2.4. Időigényes a megfelelő AV keresése

Ha már vannak előzetes ismereteink a témában, ismerjük a megfelelő, hasznosnak ítélt weboldalakat, akkor nem vesz sok időt igénybe egy adott problémához, algoritmushoz tartozó vizualizáció keresése. Egyszerűbb algoritmusoknak több vizualizációja is létezik, így sok időt vehet igénybe az is, hogy eldöntsük melyik lenne nekünk, és a diákoknak a legmegfelelőbb arra, hogy a számunkra ideális irányból, a legjobban bemutassuk azt. Ha pedig komplexebb algoritmusokat keresünk, akkor könnyen megeshet, hogy még nem is készült belőle vizualizáció, vagy csak kevés helyen, amit nehéz felkutatni.

2.5. Időigényes az AV oktatásba való integrációja

A tanároknak alaposan, minél részletesebben meg kell tervezniük az óráikat. Amikor egy teljesen új eszközt szeretnének alkalmazni, akkor annak használatát jól át kell gondolniuk. Ha már egy algoritmusnál AV-t vesznek igénybe, akkor szükségszerűen a többen is érdemes lenne. Adott esetben egy-egy vizualizáció megtalálása sok időt vehet igénybe, így felmerül a kérdés: mennyit kell foglalkozni azzal, hogy ha minden algoritmusra szeretnénk találni legalább egy megfelelőt, ami alapján be szeretnénk azt mutatni. Ezen felül probléma lehet az, hogy hiába gyűjtünk össze mindenről egy vizualizációt, ha különböző stílusúak, különbözőként mutatnak be hasonló dolgokat, akkor a diákok számára ez zavaró lehet. Felmerülhetnek még azok a kérdések is, hogy kiknek és milyen körülmények között szeretnénk felhasználni, érdemes-e mindenkinél használni, számonkérésnél mennyire játszon szerepet az AV. Ezekre a kérdésekre, még a tervezés időszakában, csoportonként választ kell adnia a tanároknak, ami főleg első alkalommal rengeteg időt vehet igénybe. Majd később óra/téma végeztével az oktatónak reflektálva a történetekre, lehet, hogy teljes mértékben át kell formálnia a döntéseit, terveztét.

2.6. Alacsony óraszám

Magyarországon jelenleg alacsony óraszámú az informatika tantárgy a hagyományos, általános tagozatú iskolákban. Ezért sokan úgy vélekednek, hogy megfelelő tanórai időmennyiség hiányában, inkább nem is használják a technológiát, vagy gyorsabban próbálnak haladni a kötelezően előírt tananyaggal.

2.7. Tanárok és AV fejlesztők gondolkodásának különbözősége

A tanároknak vannak igényeik arra, hogy melyek azok az algoritmusok, amikből szeretnének AV-t. Az AV fejlesztők pedig általában önszorgalomból csinálnak olyanokat, amikre vagy saját maguknak szükségük van, vagy amit ők jónak látnak. A tanárok által hiányolt vizualizációkat a fejlesztők nem mindig készítik el, vagy az is előfordulhat, hogy úgy alkotnak meg egyet, hogy a tanároknak az nem tetszik, vagy mert nem tudják beleilleszteni azt a saját gyűjteményükbe, vagy csak nem úgy mutatja be az algoritmust, ahogyan azt ők szeretnék.

2.8. Tanárok általi továbbfejleszthetőség lehetősége

Bizonyos esetekben szükség lehet arra, hogy a tanárok maguk is könnyedén létrehozzanak vizualizációkat, vagy továbbfejlesszenek egyet saját igényeik szerint. Ezek konstruálása a nulláról nagyon időigényes, ha pedig már létezőket szeretnének módosítani, akkor az eléggé nehézkessé tud válni (többek között a dokumentáció hiánya, hiányosságai és a nehezen érthető kód miatt). Ezenkívül akár olyan igény is felmerülhet, hogy a diákok számára is szeretnének megadni a lehetőséget újabb AV-k elkészítésére, ezáltal fejlesztve a kreativitásukat, illetve így az általuk belevitt mélyreható munka miatt azt jobban meg is értik. Ez egy rosszul kialakított környezetben, alkalmazásban szinte lehetetlen.

2.9. Eredményesség, hatékonyság kérdése

Az algoritmus vizualizáció használatának hatékonysága az oktatásban nehezen mérhető. Több kutatás is készült erről, amiből az látszik, hogy habár jobb eredményeket értek el azon diákok, akik használták, de nem jöttek ki minden esetben merőben eltérő, javuló eredmények [1]. Így a sok többletmunkát nem feltétlenül látják célravezetőnek az informatikatanárok.

2.10. Tanulók motivációja

Nem csak a tanárok és a szoftverek szemszögéből merülhetnek fel problémák, hanem a tanulók irányából is. Ha órán bemutatásra is kerülnek a vizualizációs eszközök, tanórán kívül a diákok mégsem használják ezeket tanuláshoz [5]. Egy 335 fős nemzetközi kutatásban a résztvevők negyede egyáltalán nem is próbálta ki, és csak harmada volt, aki több mint ötször használta azt [3].

3. Tanuláshatékonyság fejlesztése

3.1. Interaktivitás osztályozása

Az AV eszközöket attól függően, hogy milyen módon változtathatók, mennyire és milyen szinten kommunikálnak a vizualizáció futtatása közben a felhasználóval, különböző szintekre, osztályokra tudjuk bontani. Myller a Naps [7] által meghatározott osztályokat továbbiakkal bővítette ki, így a kibővített taxonómiát a következőféleképpen határozta meg [6] (csillaggal jelöltem a Naps által már előzőleg meghatározott pontokat):

- No viewing* (nincs megtekintés): Egyáltalán nincsen vizualizáció, csak szöveges algoritmus bemutatása. Forráskód szerepel módosíthatóság nélkül.
- Viewing* (megtekintés): Vizualizáció már létezik, de a hallgatók, csak kivetítve láthatják azt.
- Controlled viewing (ellenőrzött megtekintés): A megtekintő saját maga irányítja a vizualizációt (például: animáció gyorsasága, elemek megvizsgálása).
- Entering input (bemenet beírása): Az algoritmus bemenete módosítható a végrehajtásuk előtt.
- Responding* (válaszadás): Kérdések jelennek meg a tartalommal kapcsolatban.
- Changing* (változtatás): Lehetőség van a vizualizáció módosítására a vizualizáció alatt.
- Modifying (módosítás): A vizualizáció módosítható a megtekintése előtt, például a bemenet, és a forráskód változtatásával.
- Constructing* (összerakás): A vizualizációt interaktívan a diákok rakják össze komponensekből, mint például szöveges elemekből, geometriai alakzatokból.
- Presenting* (bemutatás): A vizualizációt bemutatják, elmagyarázzák másoknak a készítőik, így lehetőség van visszajelzésre, megbeszélésre.
- Reviewing (reagálás): A vizualizációt azért nézik meg, hogy megjegyzéseket, tanácsokat, visszajelzéseket adjanak a vizualizációra magára, vagy a programra, algoritmusra.

3.2. Eredmények, statisztikák

Sok kutatás készült már arról, hogy mely esetekben és milyen módon is lenne érdemes használni egy algoritmus vizualizációs alkalmazást, kimutatható különbséget keresve a végkimenetet tekintve. Nagyon nehezen mérhető, hogy ténylegesen is hatékonyabb a programozásoktatás az AV használatával, vagy sem, illetve sokszor nem is jöttek ki annyira eltérő eredmények, hogy ezt ki lehessen mondani. Karavirta disszertációjában [4] összevetette az eddigi kutatásokat, eredményeket a különböző (előző fejezetben már említett) osztályok szerint (Lawrence, Bryne, Grissom, Jarc, Hansen Taylor, Lauer, Laakso, Hübscher-Younger és Narayanan, Urquiza-Fuentes cikkjeiben megjelent eredményeket használta fel). A táblázatban jól látszik, hogy mely szinteket hasonlították már össze, és milyen eredmények születtek.

	megtekintés	válaszolás	változtatás	összerakás	bemutató
nincs megtekintés	(+) Lawrence (+) Bryne (+) Grissom	(+) Bryne + Grissom	+ Lawrence		
megtekintés		(+) Bryne ~ Jarc (+) Grissom (+) Taylor	+ Hansen (+) Lawrence (-) Lauer (+) Myller + Laakso	+ Hübscher-Younger és Narayanan ~ Lauer (+) Urquiza-Fuentes	
válaszolás					
változtatás				(+) Lauer	
összerakás					
bemutató					

1. táblázat: Karavirta által összeállított táblázat, amely összeveti a különböző vizsgálatok eredményeit az AV hatékonyságára nézve

Arra a megállapításra jutott, hogy minél interaktívabb egy algoritmus vizualizációs eszköz, annál hatékonyabban lehet használni azt az oktatásban, a tanulásban.

Törley disszertációjában [10] azt is megemlíti, hogy a magatartás-, figyelemzavaros diákok, azokon az órákon, melyen használtak AV-s szoftvert nem fordult elő magatartás- vagy figyelemzavar. Így, ha ilyen diákok vannak az osztályban, akkor náluk is érdemes lehet használni programozásoktatás során a módszert.

3.3. Konklúzió

Levonva a következtetéseket ebben a fejezetben összegyűjtöm azokat az elemeket, amelyek megléte esetén hatékonyak (a kimutatott eredmények alapján pedig azt is kijelenthetjük, hogy minél interaktívabbnak) mondhatunk egy vizualizációt.

Az algoritmusnak, felhasznált adatszerkezetnek legyen vizuális megjelenítése (mindeközben akár pszeudokód formájában is jelenjen meg az). Az animáció, vizualizáció a felhasználó által is léptethető legyen, illetve annak gyorsaságát is tudja állítani. Az algoritmus bemenete legyen módosítható, így több különböző, speciális esetre is végig lehet azt nézni. A futtatása közben jelenjen meg magyarázó szöveg, akár kérdés formájában. A felhasználók, diákok is tudjanak összerakni az eszköz segítségével vizualizációkat, majd annak segítségével be is tudják azt mutatni.

3.4. Példa a hatékony vizualizációs eszközre

A VisuAlgo weboldal (<https://visualgo.net/en/sorting>) igyekszik a legtöbb elemet bevinni az előbb felsoroltakból. A következő képen szöveges magyarázattal látszik, hogy az egyes funkciók, milyen módon is jelennek meg a felületen, milyen módon lettek megvalósítva.

The image shows the VisuAlgo interface for the Bubble Sort algorithm. The main area displays a bar chart representing an array of numbers: 3, 38, 5, 44, 15, 47, 36, 26, 27, 2, 46, 4, 19, 50, 48. The bars are light blue, with the bars for 15 and 47 highlighted in green. Below the chart, there are several callouts in orange speech bubbles pointing to specific UI elements:

- Vizuális megjelenítés**: Points to the bar chart.
- Utasítás magyarázata, leírása**: Points to the code block on the right.
- Algoritmus**: Points to the code block on the right.
- Bemenet módosítása**: Points to the input field at the bottom left.
- Sebesség állítása**: Points to the speed slider at the bottom left.
- Léptetés**: Points to the step-through button at the bottom center.

The code block on the right shows the following code for Bubble Sort:

```

Bubble Sort
Swapping the positions of 47 and 15.
Set swapped = true.
do
  swapped = false
  for i = 1 to indexOfLastUnsortedElement-1
    if leftElement > rightElement
      swap(leftElement, rightElement)
      swapped = true
  while swapped
  
```

At the bottom of the interface, there is a status bar showing "slow" and "fast" indicators, a progress bar, and navigation buttons. The text "About Team Terms of use" is visible in the bottom right corner.

1. ábra: VisuAlgo felülete

4. Tanulók motiváltságának javítása

4.1. Okok

Az előző részben már szóba került, hogy a hallgatók nem eléggé motiváltak az algoritmus vizualizációs oldalak használatára. Knobelsdorf [5] cikkében már vizsgálta, hogy a diákok miért is nem használják az AV-t a tanuláshoz. A következő felsorolásban (Knobelsdorf által említett indokokkal együtt) nagyobb pontokba szedve veszem sorra a lehetséges okokat:

- Az internalizációs folyamatot nem stimulálja eléggé az eszköz, a diákok csak néha veszik igénybe azt, így nem tudják használni a tanuláshoz. [5]
- Az internalizációs folyamat már megtörtént, így már nincs szükség a használatára. [5]
- Sok esetben nem próbál az eszköz kapcsolatba lépni a felhasználóval, vagy nem elég interaktív, így nem is kellőképpen köti le a diákok figyelmét.
- Kezdők számára nehézséget okozhat olyan vizualizáció használata, értelmezése, melyet már gyakorlott programozók készítettek, akik másféleképpen valósítanak meg egy vizualizációt, mint amely segítségével a diákok könnyen megértenék az adott algoritmust.
- Nem érzik motiválnak magukat, hogy vizualizációkat használjanak.
- Kevésbé tehetséges diákok nem tudják használni, értelmezni a vizualizációkat, illetve azok használatát egyedül. [3]

4.2. Tanórai használat

Érdeemes lenne a tanórákon minél többször szerepeltetni AV-s alkalmazásokat, így a diákok hozzá szoknak a használatukhoz, megértik annak jelentőségét, hasznosságát, így később más környezetben (akár otthon) is elő fogják venni az ilyen jellegű eszközöket, amelyek segítik őket a tanulásban, a könnyebb megértésben. Sőt, akár lehetne külön keretet szentelni annak, hogy megtanítsuk a diákoknak az AV-s eszközök általános használatát, különböző szempontok szerinti értékelését, hogy ezáltal szabadon kutathassanak az interneten minél jobb, részletesebb, hasznosabb vizualizációk után.

4.3. Stílus, kinézet, kommunikáció

A diákok figyelmét jobban lekötik azok a weboldalak, amelyek mai szemmel szépen néznek ki, designosak. Így tehát érdemes lenne úgy megalkotni a vizualizációkat, hogy a tanulóknak tetszen az, lekösse a figyelmüket úgy, hogy mindeközben tanulnak is. Ehhez a vizualizációk elkészítése közben a fejlesztőknek össze kell vetni azon igényeket, hogy mely szükséges elemek jelenjenek meg a felületen úgy, hogy azok kellő részletességűek legyenek, mégis az oldal átlátható maradjon, és a felhasználók figyelmét se veszítse el. Ennek eléréséhez fontos a kommunikáció a fejlesztők, a tanárok, és a UX (user experience) szakemberek között (egy kommunikációs csatorna a felületen elősegítheti ezt a folyamatot). Így a tanár igényei, a fejlesztő szerinti megvalósíthatóság, és a designer meglátásai tudnak találkozni. A kommunikációt elő lehet segíteni egy github-os repoval, vagy egy fórummal is, így bárki felvetheti ötleteit, sőt akár meg is valósíthatják azokat, amelyek ezután könnyűszerrel az éles weboldalba is belekerülhetnek.

4.4. Gamification

„A gamifikáció alatt azt a folyamatot értjük, amely során a játékokban meglévő pszichológiailag motiváló mechanizmusokat és elemeket hagyományosan nem játékos környezetbe helyezzzük, ezzel elősegítve az eredeti folyamatok sikerességének, és hatékonyságának növelését.” [9] Gamification használatával a diákok könnyebben rávehetők a tanulásra, illetve ezáltal az eszköz használatára is. Ezt többféleképpen, többféle módszerrel is megtehetjük. A megvalósíthatóság a legkönnyebb akkor lenne, ha az eszköz tulajdonképpen egy weboldal, melyet az interneten bárki elérhet. Mindenki csinálhat magának egy saját profilt, majd a tanár (afféle adminként) átláthatja a diákok előrehaladását. Érdeemes lehet pontrendszert, jutalomrendszert bevezetni, amelyekkel a diákok nemcsak visszajelzést kapnak, hanem egymással versengetnek is (ezt azonban csak kellő elővigyázatossággal, és odafigyeléssel szabad).

4.5. Példa a motiváló vizualizációs eszközre

A CodinGame honlapján (<https://www.codingame.com/home>) például jelvények, szimbólumok gyűjtése is motiválja a felhasználókat arra, hogy minél több feladványt megoldjanak, illetve ezt minél több programozási nyelven tegyék meg, vagy éppen nehezítésekkel oldjanak meg egy-egy feladatot. Összehasonlíthatjuk magunkat, versengetünk másokkal akár pontokért, akár fizikai jutalmakért. Mindeközben a vizualizációk látványvilága is tetszetős, amely próbálja megtartani az oldalra látogatókat.

The screenshot displays the CodinGame user profile page. On the left, the 'PROGRESSION' section shows a crown icon and 'Level 18', with a progress bar at '74 / 828 XP'. Below it, the 'RANKING' section shows a trophy icon and '35,670TH I'm a Rookie'. The main content area is titled 'ABOUT ME' and 'Introduce yourself to the community'. It features an 'ACHIEVEMENTS SHOWCASE' with a table of progress: Bronze (36/61), Silver (37/72), Gold (26/80), and Legend (1/48). Below this are 'Most used languages' with achievements like 'C# Addict' (+75 XP), 'C++ Lover' (+45 XP), and 'Java Explorer' (+10 XP). At the bottom are 'Best unlocked achievements' including 'Come get some' (+405 XP) and 'Semaster' (+70 XP).

2. ábra: CodinGame achievement rendszere

5. További igények a „tökéletes” algoritmus vizualizációs eszközhöz

Végül szeretnék még pár támpontot leírni arra, hogy egy algoritmus vizualizációs eszköznek milyen követelményeknek kell, hogy megfeleljen, ha azt szeretnénk, hogy minél egyszerűbb, használhatóbb, strukturáltabb legyen, amellyel aztán az oktatók feladatai csökkennének. A leírásba Törley Gábor [10] által leírt pontokat is belefűztem, melyek szerinte olyan tulajdonságok, amikkel egy jó demonstrációs eszköznek rendelkeznie kell.

Mindenki számára elérhető, könnyedén használható eszköznek kell lennie (példaként lehet egy weboldal, amihez egy böngészőn kívül nincs semmire szükség, tehát nem kell feltelepíteni hozzá semmit sem). A vizualizációk felülete egyszerű, egyértelmű legyen, akár külön leírás nélkül is meg lehessen érteni a használatukat.

Az oldalon fellelhető legyen az összes vizualizáció, amire az oktatás során szükség lehet. Így minden AV egy helyen lesz megtalálható, a tanároknak nem kell több oldalon keresniük a megfelelő vizualizációkat, és így mindegyik hasonló stílusú lesz, ezáltal nem kell a diákoknak sem a különböző irányzatokhoz hozzászokniuk.

Az alkalmazáshoz tartozzon egy részletes tervezet (tematikus tervekkel, óratervekkel évfolyamonként), mely jelentősen csökkentené az oktatók terheit, ezáltal magabiztosabban, könnyebben fel tudnák használni az eszközt az óráikon. Ha a nagyon alacsony óraszám nem teszi lehetővé, hogy tanórán fel lehessen használni, akkor akár egy olyan tervezet is, mellyel a diákok (viszonylag kevés tanári segítséggel) maguktól, otthoni munkaként is végig tudnának menni az anyagrészekben. Gondolok itt arra, hogy az órán bemutatásra kerülnek az algoritmusok, majd otthon a tanulók gyakorlásként vizuálisan is végig tudják nézni a működésüket.

Az eszköz legyen jól megírva, moduláris legyen, és legyen részletesen dokumentálva. A kód nyílt forráskódú legyen, hogy ezáltal aki később szeretne újabb vizualizációkat, az könnyedén létre tudjon hozni. Így a tanároknak, diákoknak nem csak az a szabadsága lesz meg, hogy bármikor létrehozhatnak ilyeneket könnyedén, hanem hogy azok hasonló stílusúak, hasonló kinézetűek lesznek, mint a már elkészült régebbi honlapon lévő művek.

Irodalom

1. Grissom, S., McNally, M. F., Naps, T. L. (2003). *Algorithm visualization in CS education: Comparing levels of student engagement*. Proceedings of the 2003 ACM Symposium on Software Visualization, SoftVis '03, San Diego, California, ACM Press, pp87-94, 2003.
2. Christophet D. Hundhausen, Sarah A. Douglas Andjohn T. Stasko: *A Meta-Study of Algorithm Visualization Effectiveness*. Journal of Visual Languages and Computing, pp259-290, 2002.
3. Essi Lahtinen, Hannu-Matti Järvinen, Suvi Melakovski-Vistbacka: *Targeting Program Visualizations*. Proceedings of the 12th Annual Conference on Innovation & Technology in Computer Science Education, pp256-260, 2007.
4. Ville Karavirta: *Facilitating Algorithm Visualization Creation and Adoption in Education*. Helsinki University of Technology, 2009.
5. Maria Knobelsdorf, Essi Isohanni, Josh Tenenber: *The Reasons Might be Different – Why Students and Teachers Don't Use Visualization Tools*. Proceedings of the 12th Koli Calling International Conference on Computing Education Research, pp1-10, 2012.
6. Niko Myller, Roman Bednarik, Erkki Sutinen, Morechai Ben-Ari: *Extending the engagement taxonomy: Software visualization and collaborative learning*. ACM Transactions on Computing Education, New York, USA, 2009.
7. Thomas L. Naps, G. Rössling, V. Almstrum, W. Dann, R. Fleischer, C. Hundhausen, A. Korhonen, L. Malmi, M. McNally, S. Rodger, J. Á. Velázquez-Iturbide: *Exploring the role of visualization and engagement in computer science education*. ITiCSE on Innovation and Technology in Computer Science Education (ITiCSE'02). ACM Press, pp131–152, 2002.
8. Guido Rössling, J. Angel Velázquez-Iturbide: *Editorial: Program and Algorithm Visualization in Education*. ACM Transactions on Computing Education (TOCE) - Special Issue on the 5th Program Visualization Workshop, New York, USA, 2009.
9. Schuchmann Júlia: *A közösségfejlesztés új útjai*, Kodolányi János Főiskola, 2016.
10. Törley Gábor: *Vizualizáció a Programozástanításban*, ELTE, 2013.
11. Végh Ladislav: *A Programozás Tanulásának és Tanításának Támogatása Elektronikus Tananyagba beépíthető Interaktív Animációs Modellekkel*, ELTE, 2017.