Algorithm Visualization Environments: Degree of interactivity as an influence on student-learning

Pálma Rozalia Osztián
Informatics Department
Sapientia Hungarian University of Transylvania
Targu Mures, Romania
osztian.palma@ms.sapientia.ro

Zoltán Kátai
Informatics Department
Sapientia Hungarian University of Transylvania
Targu Mures, Romania
katai_zoltan@ms.sapientia.ro

Erika Osztián
Informatics Department
Sapientia Hungarian University of Transylvania
Targu Mures, Romania
osztian@ms.sapientia.ro

***Abstract* — Nowadays, online learning environments have become very popular for teaching algorithms. To improve the learning process these environments often include various visualizations of the algorithms. Because of the abstract nature of algorithms, expressive animations which illustrate these processes have become critically important educational tools. Another important feature of online learning environments is to what extent the user should be involved in the learning process. Previous research in this field has produced mixed results. While some studies emphasize that there is a benefit to user control, others conclude that interrupting the animation process can also have negative effects. Taking into account this previous work we have used a novel online learning tool (AlgoRythmics) which includes visualizations of ten basic computer algorithms (searching and sorting strategies). In this environment three levels of interactivity have been defined: no-interactivity (passive viewing of instructional material: students are independent observers), half-interactivity (students are partially involved: at specific key moments the animation suddenly stops, and user interaction is required) and full-interactivity (full control is given to users: students are invited to orchestrate the algorithm by predicting the entire operation sequence for a given input). We focused on the AlgoRythmics illustration of the Shell sort algorithm and, more specifically, on the influence that the degree of interactivity has on students' learning.**

We examined 137 first year undergraduate students (14% females) who were divided in three "equivalent" subgroups based on their prior knowledge (0, 1-3 or 4 years of programming during high school studies). We chose the Shell sort algorithm since none of the participants were familiar with this sorting strategy. The three instructional conditions assigned to the three subgroups were: no-, half- and full-interactivity. Our results show that there are no significant differences between students' performance from the three specific subgroups. This finding is in line with some previous research results that emphasize that each interactivity level might have both positive and negative effects. Since students' learning styles are different, an important characteristic of online learning environments should be that they provide the possibility for everyone to choose the most appropriated interactivity level.

Keywords — interactivity, algorithms, animation, e-learning

I. Introduction

Computational thinking can be considered one of the most important and fundamental abilities for people from the XXI. century [23]. Introducing computer based algorithms in education can improve not only students' programming skills, but also their computational thinking [6]. According to Turing [21], understanding how computer algorithms work assumes that students can imagine *"a clear mental picture of the state of the machine at each moment in the computation"*. Since computer algorithms are abstract dynamic processes, animations have become the most common tools for visualizing them.

Berney and Bétrancourt [2] emphasize that building a coherent mental model from animations could largely be influenced by learners' individual characteristics (for example prior knowledge). Other critically important moderating factors could be the delivery features of instructional material and the characteristics of learning tasks. In this study we focus mostly on how student engagement in algorithm visualization (AV) processes might influence the learning outcome.

The relevant research in the field of AV reports mixed results regarding their educational value. One of the most important studies on AV effectiveness [10] concludes that the way students use visualizations is more important than the visualizations themselves. According to Shaffer et al. [20], an important conclusion from the literature is that to make AVs pedagogically useful, they must support student interaction and active learning.

We implemented our study in the AlgoRythmics [8] learning environment that incorporates dance choreography illustrations and interactive abstract animations for ten computer algorithms. The animations include the so-called interactive prediction feature on three levels: *"no-interactivity"* (passive viewing), *"half-interactivity"* (the animation stops at predefined key moments and users have to indicate the next movement/ operation) and *"full-interactivity"* (users have to orchestrate the algorithm by interactively predicting the entire operation sequence). Accordingly, our investigation focused on the influence that varying degrees of interactivity have upon students' learning. We analyzed this issue with respect to the Shell sort algorithm. The experiment was conducted with first year undergraduate students with different levels of prior knowledge in programming.

II. DIFFERENT LEVELS OF ENGAGEMENT WITH ALGORITHM VISUALIZATIONS

Interactive visualization has been employed in computer science education since the 1980s. In 2002 Naps et al. [16] reported a strong perception among educators that visualization can help (almost all respondents for their survey agreed with the statement *"Using visualizations can help learners learn computing concepts"*). Interestingly and contradictory, in the same year a meta-analysis [10] reported mixed results regarding the pedagogical benefits of visualization technology. However, Naps and his co-authors re-analyzed the twenty-one experiments included in the meta-analysis performed by Hundhausen et al. [10] and noticed that ten out of twelve (83%) experiments using learner engagement yielded a significant result. On the other hand, in the case of those nine experiments that manipulated representations, only three (33%) produced a significant result. Based on this observation, Naps et al. [16] state that these results suggest that what learners do, not what they see, may have the greatest impact on learning. In other words, they conclude that AV technology is of little educational value unless it engages learners in an active learning activity.

In addition, Naps et al. [16] propose an engagement taxonomy including six different forms of learner engagement with AV technology: 1) no viewing, 2) viewing (student passively views an AV), 3) responding (student responds to questions about the content while viewing an AV), 4) changing (student changes the AV by, for example, providing input data to the algorithm), 5) constructing (student constructs an AV), 6) presenting (student presents an AV to others).

In the study performed by Grissom, McNally and Naps [7] the authors compare the performance of three treatment groups having different levels of engagement with visualizations: *no viewing* (not seeing any visualization), *viewing* (simply viewing visualizations for a short period in the classroom), and *responding* (interacting directly with the visualizations for an extended period outside of the classroom). These authors also conclude that learning increases as the level of student engagement increases.

In the present study we have included three levels of engagement: *viewing*, *responding*, and *constructing*. To implement the responding level, we applied the interactive prediction method. Regarding the constructing level of engagement Karavirta and Shaffer [12] note that a variation on this approach gives a data structure and an algorithm, and expects the student to simulate the algorithm. In other words, students are invited to imitate the steps of an algorithm by manipulating the interface to control the progress of the AV. In the following we will identify these three conditions as: *no-interactivity*, *half-interactivity*, and *full-interactivity*.

Studying learning, with dynamic visualizations that incorporate different interactivity levels, is a multifaceted research area including intertwined factors. For example, two important principles that relate to the interaction design of dynamic representations are learner control-segmenting and learner control-pacing [17]. It can be observed that *half-interactivity* implicitly generates segmentation and *full-interactivity* implicitly results in learner control of pacing. In addition, from a learning perspective, we can distinguish between functional interactivity and cognitive interactivity [17]. While an investigation focuses on cognitive interactivity aspects, the analyzed condition might differ from a functional interactivity perspective. More generally, the presence of a combined effect of several interdisciplinary factors (psychological factors, animation design, learning environment, didactically implementation, etc.) could offer a plausible explanation as to why results in this field of research are contradictory (as mentioned above).

III. THE ALGORYTHMICS ONLINE LEARNING ENVIRONMENT

The AlgoRythmics environment, where our investigation was implemented, is built around five learning steps (video, animation, in control, code building, code comes alive) and includes all six levels of engagement. For example, the animations function in three modes: *viewing*, *responding* and *constructing*. In *viewing* mode (*no-interactivity*) students are independent observers and the following basic user options are available to them: play, pause and stop buttons and an animation speed slider "Fig. 1".

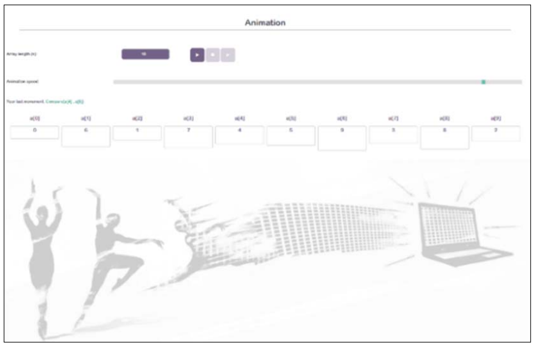

Fig. 1. Viewing mode (no-interactivity)

In *responding* mode (*half-interactivity*) students are partially involved which means that at predefined key moments the animation suddenly stops and user interaction is needed. In *constructing* mode (*full-interactivity*) learners need to orchestrate the whole animation process, they are the constructors of the algorithm, the embodiment of the operations. In order to choose the next correct movement, the following user options are available in both *responding* and *constructing* modes: select the pair of elements, compare and swap "Fig. 2". In the event they are unsure how to proceed, the online learning environment provides the opportunity to ask for help (hint).

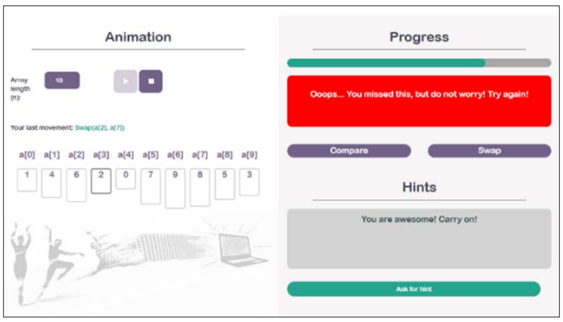

Fig. 2. Responding (half-interactivity) and constructing (full-interactivity) mode

We allocated each interactivity level to a specific group: G_1 (*no-interactivity*), G_2 (*half-interactivity*) and G_3 (*full-interactivity)*.

In our experiment all participants were able to complete each step once, we did not use changing abilities of the learning environment, thus in case of each AV the input was a predefined sequence of numbers given by a teacher.

IV. Research Questions

Based on a review of the literature we hypothesized that learning would increase as the level of student engagement increases.

More precisely and additionally, we addressed the following research questions:

1) RQ1: Does increased student engagement benefit the learning outcome (results in higher post-test scores)?

2) RQ2: How does students' prior programming knowledge influence comprehension?

3) RQ3: Is there a significant difference between female and male students' performance?

4) RQ4: Is there a correlation between the level of interactivity and the nature of the acquired knowledge?

V. Method

We designed a three phase experiment (*pre-test*, *learning phase* and *post-test*) and the investigation took place at an Eastern Europe University at the beginning of the 2019/2020 academic year.

A. Participants

In our research 137 students were recruited, 3 did not complete the preliminary questionnaire correctly, resulting in 134 study participants (14% females). All participants were first year undergraduate students from five different educational programs: Informatics, Computer Science, Automation, Mechatronics and Mechanical Engineering. Participants were assigned to 1 of 3 categories based on their prior programming experience: No-prior knowledge *(NP: 0 years of high school programming experience)*, Basic-prior knowledge *(BP: 1,2 or 3 years of high school programming experience in natural science classes; one to two classes per week)* and High-prior knowledge *(HP: 4 years of high school programming experience in informatics classes; five to seven classes per week)*. Students from each category were randomly assigned to the three groups G_1, G_2 and G_3.

During the *learning-phase* and *post-test* 46 students served in G_1 group and 44 students served in both G_2 and G_3 groups (the three eliminated participants belonged to groups G_2 and G_3).

The mean proportion of females was 10.86%, 18.18%, 13.63% yielding to a non-significant difference based on Chi-Square test ($p = 0.6 > 0.05$). The mean of prior programming knowledge was 2.17, 2.11, 2.20 (years) for the specific groups (G_1, G_2 and G_3).

Testing this proportion, we came to the same result: no significant differences (Chi-Square: $p = 0.98 > 0.05$) (see TABLE 1).

TABLE 1. Proportion of students by prior programming knowledge and gender

Total: 134 students		G_1	G_2	G_3
Gender	Male	41	36	38
	Female	5	8	6
Prior programming knowledge	NP	14	13	19
	BP	15	11	18
	HP	13	13	18
Mean of prior programming knowledge		2.17	2.11	2.20

B. Materials

The preliminary questionnaire, the *pre-test* and the *post-test* were conducted in the Socrative online classroom application. The preliminary questionnaire consisted of 8 questions: 1 acknowledgement of privacy policy, 2 questions about their demographic information, 5 questions about their prior programming knowledge (How many years had they studied informatics during high school? Are they familiar with the following sorting algorithms: bubble-, selection-, insertion-, and shell sort?).

In order to test participants' computational thinking the *pre-test* included 8 questions based on programming free tasks selected from the website of the Bebras contest [9].

The *learning phase* was built around the AlgoRythmics animation of the shell sort algorithm (on 10-length sequence), since only 7% of the participants had heard about this sorting strategy and none of them had studied it. The experiment included the animation in all the three modes: *viewing*, *responding* and *constructing*.

The *post-test* questionnaire consisted of the following 12 questions (potential range: 0-12):

- $Q_{1\text{-}6}$: Considering a "3-1 Shell sort strategy" (for gap-values 3 and 1) on the sequence x[0..6] = {1,19,7,8,12,11,9}, which are the first three steps (indicate the operation and the pair of elements to be swapped or compared)? FIRST/ SECOND/ THIRD (x[?], x[?])
- $Q_{7\text{-}8}$: Consider the "3-1 Shell sort strategy" for the sequence x[0..6] = {1,19,7,8,12,11,9}. After the compare(x[3], x[6])/ compare(x[5],x[6]) operation which is the next pair of elements to be compared?
- $Q_{9\text{-}12}$ (generalization): How many compare/swap operations are performed by a "3-1 Shell sort" algorithm for a 7-length increasing/ decreasing ordered sequence?

C. Procedure

The experiment lasted a total of 2 hours. The *pre-test* took 30 minutes, which was held for all three groups together in an amphitheater of the university. As a preparation for the *learning phase* we presented the AlgoRythmics online learning tool because many of the students were encountering this online learning environment for the first time. The quick demonstration of the learning tool was presented with the insertion sort algorithm. During the 15-minute presentation (at which all the participants were present) it was demonstrated how animation works in all three modes (*no-*, *half-* and *full-interactivity*).

After this each group (G_1, G_2 and G_3) was assigned to a computer laboratory where the *learning phase* began. During this, students were required to register for the online learning environment and after this they started to complete the designated course, which took approximately 30 minutes (this amount of time was set with regard to the group G3; the other two groups, of course, completed their learning task sooner). For each group the learning phase included two viewings of the animation. As a first step, all groups watched the animation in *viewing* mode (the core form of engagement, [16]). During the second viewing, groups G_1, G_2 and G_3 watched the animation with *no-*, *half-* and *full-interactivity*, respectively.

At the end of the experiment participants from all groups were invited to answer the *post-test* questions. This phase took approximately 20 minutes.

VI. Results and Discussion

Statistical analysis was performed using SPSS statistical software. The *pre-test* performances (see "Fig. 3") were analyzed with a one-way Analysis of Variance (ANOVA). The independent variable was the instructional condition (*no-, half-, full-interactivity*), and the dependent variable was the pre-test score (Levene's test showed that equal variances could be assumed: $p = 0.93 > 0.05$). As we expected, no significant differences were found ($F(2, 130) = 0.007, p = 0.99 > 0.05$).

As a next step, we analyzed participants' *post-test* performance (see "Fig. 3") with a one-way Analysis of Covariance (ANCOVA). Again, the independent variable was the instructional condition, and the dependent variable was the post-test score (the assumption of homogeneity was met; $p = 0.27 > 0.05$). The *pre-test* performance was used as a covariate. Although a moderate decrease was noticed (G_1: 56%, G_2: 53%, G_3: 51%) the differences were not significant ($F(2, 130) = 0.846, p = 0.432 > 0.05$).

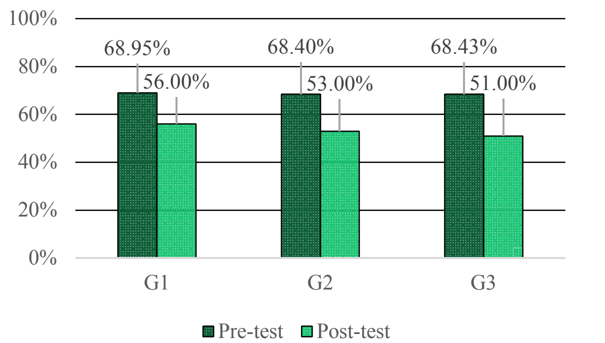

Fig. 3. Pre-test and post-test results based on the level of interactivity

This result does not support the assumption that learning will increase as the level of student engagement increases. In addition, it suggests that a universally optimal interactivity level cannot be established. Clearly, this conclusion is contrary to the majority of prior research in this field. For example, Shaffer et al. [20] conclude (based on [10] and [16]) that a growing body of evidence indicates that the most important factor that contributes to the effectiveness of learning with AV appears to be engagement of the students' attention.

On the other hand, some previous results harmonize with our finding. For example, Jarc, Feldman, and Heller [11] also found no significant difference in educational outcomes for students who used the interactive version of their AV system (interactive prediction) compared to their peers who used the *non-interactive* version (*passive viewing*). Interestingly, Shaffer et al. [20] use the expression of *"engagement of the students' attention"*. It seems that watching an AV could be engaging even without any interactivity. In addition, Myller, Laakso and Korhonen [15] also report that they cannot confirm their hypothesis that ET (engagement taxonomy: *viewing* and *changing* levels) level would have a significant effect on the learning outcomes.

Similar results were found regarding the pacing functionalities of the animations. Prior research in this field also reported inconsistent findings. While some studies showed a benefit of control ([3], [14], [19]), others found no benefit [1]. One of the most important findings of the meta-analysis performed by Berney and Bétrancourt [2] is that the positive effect of animation over static graphics was found only when learners did not control the pace of the display.

Our results suggest that all three levels of engagement we analyzed may have both advantages and disadvantages. For example, a possible explanation of why interactivity may not always guide students to better learning outcomes would be that interrupting the animation process may prevent students from immersing themselves deeply in the visualization process. Comprehending an algorithm assumes that learners are able to construct an overall picture (strategy of the algorithm) from small elements (individual steps of the algorithm). Fragmenting the visualization could obstruct students in this building process. Boucheix et al. [4] came to a similar conclusion regarding visual cueing. Their study demonstrated that cueing could counteract the spontaneous exploration of the animation.

Another contributing factor to this result could be that the increase in student engagement generated an increase in usability difficulty too. As mentioned above, although we focused on the cognitive functionality aspects, the conditions differed from the perspective of functional interactivity too. We tried to diminish this influence by including that 15 minute presentation where it was demonstrated how animation works, especially in the interactive conditions. This aspect was also taken into account when setting the amount of time allocated to this phase of the experiment.

In addition, with respect to the interactive prediction method Jarc et al. [11] noticed that sometimes students view the integrated questions as a guessing game and they are not taking it seriously. And, as a consequence, the objective of engaging students' attention is not reached.

A. Results grouped by prior programming experience

We also analyzed participants' *pre-* and *post-test* performances grouped according to their prior programming experience (see "Fig. 4"; independent variable: NP, BP or HP; dependent variable: *pre-test*/ *post-*

test score). In both cases, as we expected, the ANOVA test showed significant differences (*pre-test:* $F(2, 130) = 3.455$, $p = 0.034 < 0.05$; *post-test:* $F(2, 130) = 11.299$ $p = 0.00 < 0.05$). After performing further analysis (planned contrasts) we observed that ("<" denotes not significant increase, "<<" denotes marginally significant increase and "<<<" denotes significant increase):

- Pre-test: NP << BP < HP;
- Post-test: NP < BP <<< HP;

Interestingly, while the differences between NP and HP students were almost equal in both cases (*pre-test:* 60% vs. 74%, *post-test:* 46% vs. 61%), BP students' *pre-test* performance was closer to their HP colleagues' performance and their *post-test* performance closer to NP students' performance. A possible reason could be that 66% of the post-test questions (Q_{1-8}) focused on the operation sequence the algorithm generates, while 33% of them (Q_{9-12}) related to algorithm efficiency issues, a concept only included in the HP students' high school curriculum.

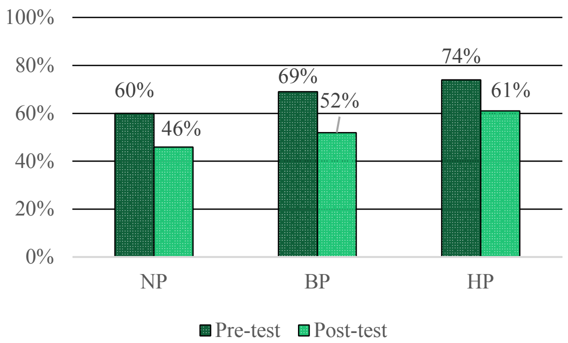

Fig. 4. Pre- and post-test results based on prior programming knowledge

In order to investigate whether the different category of participants (NP, BP, HP) performed equally well at each interactivity level a two way ANOVA was conducted. The two independent variables were the instructional condition (*no-*, *half-*, *full-interactivity*) and students' prior programming experience level (NP, BP, HP), and the dependent variable was students' *post-test* score (Levene's test showed that the variances were equal: $p = 1.54 > 0.05$). No interaction was detected ($p = 0.6 > 0.05$). When we applied three distinct ANOVA tests for each category of students (NP, BP, HP), we also found no significant differences (see "Fig. 5").

This result is consistent with some previous research. For example, Myller, Laakso and Korhonen [15] divided their subjects into two groups based on their pre-test results: NPK (with no prior knowledge in the topic), SPK (with prior knowledge in the topic). Although these authors emphasize that NPK students benefited more from using the visualization on higher engagement level, they admit that the differences between the two groups are not statistically significant.

On the other hand, according to the task appropriateness principle [17], interactive dynamic visualizations may reduce cognitive load in tasks that requires high mental effort. However, they may also inhibit processing by providing unnecessary support [18]. Since (1) at the beginning of the learning phase all participants watched the animation in viewing mode and (2) participants with prior knowledge in programing had already studied the insertion sort algorithm before (the Shell sort strategy is based on this algorithm), these factors could also contribute to this result.

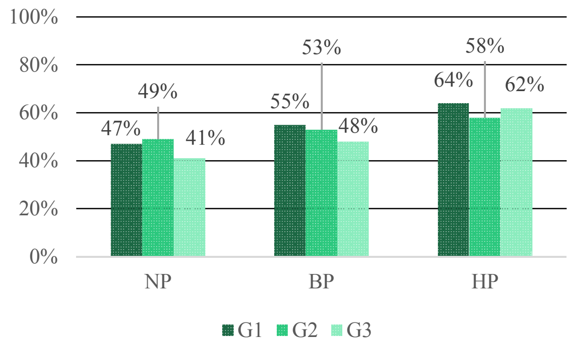

Fig. 5. Post-test results by prior programming knowledge and the level of interactivity

B. *Relation between the level of interactivity and the nature of acquired knowledge*

We designed the post-test question sequence in such a way as to have three different types of items. Questions 1-6 related to a sequence of consecutive operations (first three operations). We assumed that these kinds of question would be particularly accessible to those participants who were assigned to the *full-interactivity* condition (they had to reconstruct the entire algorithm, step by step: select the right pair of elements and apply the correct operation, again and again). Questions 7-8 focused on sporadically selected operations from the step-sequence of the algorithm. These types of questions harmonized mostly with the *half-interactivity* condition (interactive prediction). Questions 9-12 related to algorithm complexity issues and assumed that students had managed to put together an overall picture of the algorithm.

We believed that the smooth observation (*no-interactivity* condition) of the visualization would rather contribute to this overall picture.

Students' overall performance decreased as they advanced with the post-test question sequence (Q_{1-6}: 59%; Q_{7-8}: 56%; Q_{9-12}: 45%). While Q_{1-6} and Q_{7-8} scores were close to each other, the Q_{9-12} score was significantly lower than these ones. Since questions 9-12 addressed algorithm complexity issues, this is a quite evident result. This is a plausible result from the perspective of Bloom's Taxonomy of Cognitive Development [13] too. While questions 1-8 belonged to the application level, questions 9-12 represented the analysis level.

Examining the groups separately (see "Fig. 6") the only (marginally) significant result was found when we compared the Q_{9-12} score of group G_3 with the corresponding scores of groups G_1 and G_2 (ANOVA, contrast values (-1,-1,2); $p = 0.058$). A possible reason why students from group G_3 scored significantly lower on the Q_{9-12} items than their colleagues from the other two groups might be that in the *full-interactivity* condition students faced the algorithm in a very fragmented way.

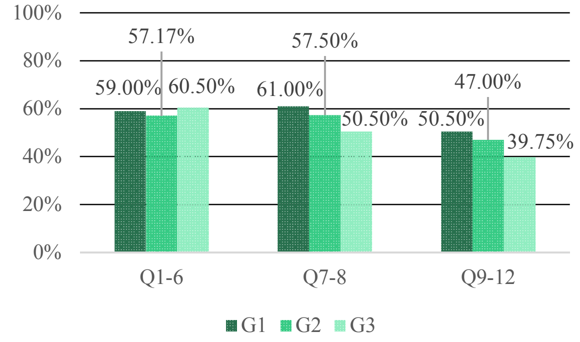

Fig. 6. Results by question type and the level of interactivity

As a next step we repeated the above analysis for each category of students (NP, BP, HP). The only significant result was found with respect to BP students. When we examined the groups (G_1, G_2, G_3) separately, we noticed interesting differences (see "Fig. 7"). Although in the case of both group G_1 and group G_3 the performances linearly decreased, the slope of decrease differed (G_1: 61.5%, 53.5%, 46%; G_3: 61.5%, 46%, 28.8%). In addition, participants from group G_2 (*half-interactivity*; interactive prediction) performed best on questions 7-8 and they reached the highest scores on these questions. These results suggest a correlation between the level of interactivity (with which the student studies the algorithm) and the nature of acquired knowledge.

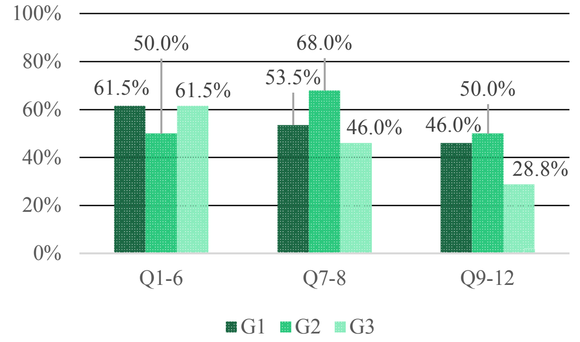

Fig. 7. Results by question type and the level of interactivity – BP students

C. Results grouped by gender

Since the most effective method for male and for female could differ, we also analyzed students' *post-test* performance grouped by gender (see "Fig. 8"). We found no significant differences. In order to investigate whether the gender groups performed equally well at each interactivity level a two-way ANOVA was conducted. The two independent variables were the instructional condition (*no-*, *half-*, *full-interactivity*) and the gender (male, female), and the dependent variable was students' *post-test* score (Levene's test showed that the variances were equal: $p = 0.09 > 0.05$). No interaction was detected ($p = 0.8 > 0.05$). Despite this fact, we observed that while males performed better than females in the *no-interactivity* condition, this relationship was reversed in the other two conditions (*half-* and *full-interactivity*).

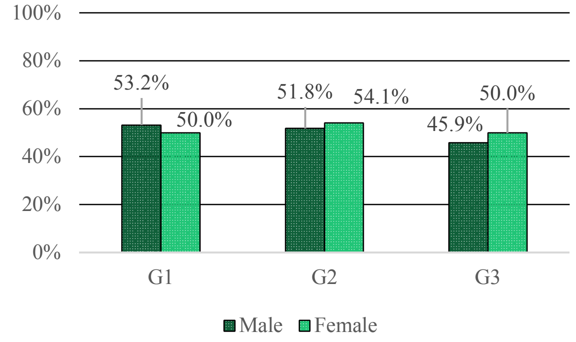

Fig. 8. Post-test results grouped by gender and the interactivity level

D. Most preferred course variant

After the *post-test*, an extra question asked participants to indicate their most preferred interactivity level (the most preferred course variant, regardless of which they were assigned to). As can be discerned from "Fig. 9" in the case of group G_1 76.09% of the students wanted one level more (G_2) and 21.74% two levels more (G_3) interactivity. Likewise, 34.09% of the students from group G_2 wanted one level more (G_3) interactivity and 59.09% were pleased with the assigned course (G_2) which was associated with *half-interactivity*. In the case of the group G_3 the majority of students (52.27%) were pleased with this course.

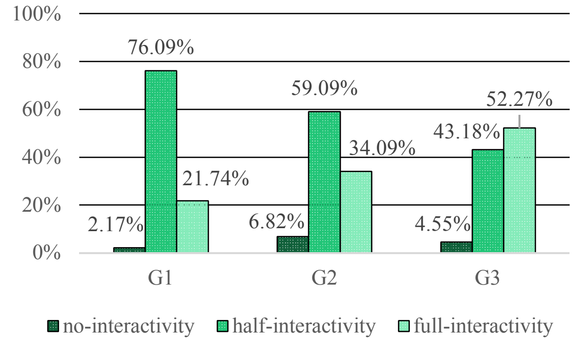

Fig. 9. Students' satisfaction level diagram

We encoded participants' choices as follows (see "TABLE 2" and "Fig. 10"):

- 0: for those participants who were pleased with the course assigned to them
- -1 or -2: for those who would prefer less interactivity (with one or two levels less)
- +1 or +2: for those who would prefer more interactivity (with one or two levels more).

TABLE 2. STUDENTS' SATISFACTION LEVEL

Assigned course	-2	-1	0	+1	+2	Avg
G_1	-	-	1 (G_1)	35 (G_2)	10 (G_3)	1.19
G_2	-	3 (G_1)	26 (G_2)	15 (G_3)	-	0.27
G_3	2 (G_1)	19 (G_2)	23 (G_3)	-	-	- 0.52
Overall average						0.31

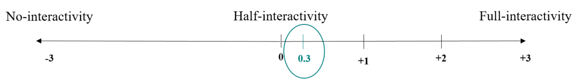

Fig. 10. Students' satisfaction level axis

Analysis of the students' responses, revealed a 0.31 mean value. This positive value suggests that participants voted for more interactivity. As shown in "Fig. 11" *half-interactivity* engagement level is closest to students' preferences. This is encouraging since several prior studies concluded in favor of this type of engagement. In an experiment implemented by researchers Byrne, Catrambone and Stasko [5], video was used as a material of the learning phase. The video stopped at certain predetermined key moments and students had to indicate what the next step would be. Grissom, McNally and Naps [7] also report that students who responded to questions integrated into the AV tool during their exploration of an algorithm showed the most improvement between a *pre-test* and *post-test*.

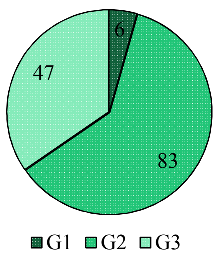

Fig. 11. Most preferred course variant

Interestingly, just as Naps et al. [16] reported a strong perception among educators that visualization can help, (although previous research result did not supported this perception), we detected a similar phenomenon with respect to learners: they voted for more interactivity, despite the fact that their performance did not provide evidence in this sense.

On the other hand, we should emphasize that students' satisfaction choice might be based on their imagination and not on real experience. In future research we plan to re-analyze this apparently contradictory result (usually, if students are interested in an approach, they will be more likely to be involved more actively, which results in more effective learning).

VII. Limitations

One of the limitations of this study is that we included in our investigation only one algorithm: the AlgoRythmics animation of the Shell sort algorithm.

Another limitation could be that participants were assigned to a group. A further research issue could be as follows: what if students could freely choose the group and control their learning (while the system records interactive steps)?

With respect to the fact that differences in the level of engagement could generate, as a side-effect, differences in usability factors, future investigations would test this possible influence, for example, by an extra post-test questionnaire.

VIII. Conclusions

This study provides a new insight into the research field of AV and valuable guidelines for instructional design. Based on the engagement taxonomy introduced by Naps et al. [16] we examined three interactivity levels: *viewing*, *responding* and *constructing*. We implemented these levels of engagement in the AlgoRythmics environment (using the animation that illustrates the Shell sort algorithm). We identified the designed instructional conditions as: *no-interactivity* (passive viewing), *half-interactivity* (interactive prediction) and *full-interactivity* (algorithm orchestration). One of the most important conclusions of our investigation is that a universally optimal interactivity level cannot be established.

Based on this study, we cannot confirm the hypothesis we formulated regarding the AlgoRythmics visualization of the Shell sort algorithm - that the level of engagement would have a significant effect on the learning outcomes. Our findings suggest that all three levels of engagement that we analyzed may have both advantages and disadvantages. Interestingly, Urquiza-Fuentes and Velázquez-Iturbide [22], after examining several previous research articles in the field of AV effectiveness, also concluded that improvements in knowledge acquisition had been detected at any engagement level.

We also found that students with different prior experience can equally benefit from each engagement level. Additionally, some of our results suggest a correlation between the level of interactivity and the nature of acquired knowledge.

Since we analyzed only one specific algorithm visualization, it is clear that in order to generalize these findings, further research is needed that also addresses such possible influencing factors as algorithm complexity, usability issues, design parameter, the learning environment, etc.

Everybody deserves to learn and thus everybody is different, the most effective learning style should differ in the case of each student. Therefore, we can declare that all learning environments should be able to present AV with different interactivity levels. If this can be achieved, all participants would find the best learning style for themselves and this could only lead to better results.

Acknowledgment

The authors wish to acknowledge the help of those colleagues who supported them in developing the AlgoRythmics environment. The authors would like to thank all subjects for participating in this research.

References

[1] Adesope, O. O., & Nesbit, J. C. (2012). Verbal redundancy in multimedia learning environments: A meta-analysis. Journal of Educational Psychology, 104(1), 250.

[2] Berney, S., & Bétrancourt, M. (2016). Does animation enhance learning? A meta-analysis. Computers & Education, 101, 150-167.

[3] Boucheix, J. M., & Guignard, H. (2005). What animated illustrations conditions can improve technical document comprehension in young students? Format, signaling and control of the presentation. European Journal of Psychology of Education, 20(4), 369-388.

[4] Boucheix, J. M., Lowe, R. K., Putri, D. K., & Groff, J. (2013). Cueing animations: Dynamic signaling aids information extraction and comprehension. Learning and Instruction, 25, 71-84.

[5] Byrne, M. D., Catrambone, R., & Stasko, J. T. (1999). Evaluating animations as student aids in learning computer algorithms. Computers & education, 33(4), 253-278.

[6] Denning, P. J. (2017). Remaining trouble spots with computational thinking. Communications of the ACM, 60(6), 33-39.

[7] 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).

[8] https://algorythmics.ms.sapientia.ro/

[9] https://www.bebras.org/

[10] Hundhausen, C. D., Douglas, S. A., & Stasko, J. T. (2002). A meta-study of algorithm visualization effectiveness. Journal of Visual Languages & Computing, 13(3), 259-290.

[11] Jarc, D. J., Feldman, M. B., & Heller, R. S. (2000). Assessing the benefits of interactive prediction using web-based algorithm animation courseware. ACM SIGCSE Bulletin, 32(1), 377-381.

[12] Karavirta, V., & Shaffer, C. A. (2015). Creating engaging online learning material with the JSAV javascript algorithm visualization library. IEEE Transactions on Learning Technologies, 9(2), 171-183.

[13] Krathwohl, D. R. (2002). A revision of Bloom's taxonomy: An overview. Theory into practice, 41(4), 212-218.

[14] Mayer, R. E., & Chandler, P. (2001). When learning is just a click away: Does simple user interaction foster deeper understanding of multimedia messages?. Journal of educational psychology, 93(2), 390.

[15] Myller, N., Laakso, M., & Korhonen, A. (2007, June). Analyzing engagement taxonomy in collaborative algorithm visualization. In Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education (pp. 251-255).

[16] Naps, T. L., Rößling, G., Almstrum, V., Dann, W., Fleischer, R., Hundhausen, C., ... & Velázquez-Iturbide, J. Á. (2002). Exploring the role of visualization and engagement in computer science education. In Working group reports from ITiCSE on Innovation and technology in computer science education (pp. 131-152).

[17] Plass, J. L., Homer, B. D., & Hayward, E. O. (2009). Design factors for educationally effective animations and simulations. Journal of Computing in Higher Education, 21(1), 31-61.

[18] Schnotz, W., & Rasch, T. (2005). Enabling, facilitating, and inhibiting effects of animations in multimedia learning: Why reduction of cognitive load can have negative results on learning. Educational Technology Research and Development, 53(3), 47.

[19] Schwan, S., & Riempp, R. (2004). The cognitive benefits of interactive videos: learning to tie nautical knots. Learning and instruction, 14(3), 293-305.

[20] 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), 1-22.

[21] Turing, A. M. (2009). Computing machinery and intelligence. In Parsing the Turing Test (pp. 23-65). Springer, Dordrecht.

[22] Urquiza-Fuentes, J., & Velázquez-Iturbide, J. Á. (2009). A survey of successful evaluations of program visualization and algorithm animation systems. ACM Transactions on Computing Education (TOCE), 9(2), 1-21.

[23] Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.