CREATING ENGAGING EXERCISES WITH MOBILE RESPONSE SYSTEM (MRS)

Debzani Deb
Dept. of Computer Science
Winston-Salem State University
Winston-Salem, NC, USA
1-336-7502496
debd@wssu.edu

M. Muztaba Fuad
Dept. of Computer Science
Winston-Salem State University
Winston-Salem, NC, USA
1-336-7503325
fuadmo@wssu.edu

Mallek Kanan
Dept. of Computer Science
Winston-Salem State University
Winston-Salem, NC, USA
1-336-7502480
mkanan113@rams.wssu.edu

ABSTRACT

Computer Science instructors have been exploiting learning technology such as Algorithm Visualization (AV) for last few years to explain hard-to-understand algorithms to the learners through simulations and animations. In this work, we explore an active and highly engaging approach, namely, the construction of visualizations of the algorithms under study. Our approach is further augmented with automated assessment of students' in-class construction activities, which they execute as apps in their mobile devices. In this paper, we utilize case study, a step-by-step visualization of a construction exercise app, to explain how technology is leveraged to provide a richer way for learners to interact with a problem, and how instructor can acquire real-time evidence of learners' comprehension of covered lecture material. Our experimental evaluation shows the educational benefits of the proposed approach in terms of enhanced student learning, reduced drop-out rate and increased student satisfaction.

Keywords

Algorithm visualization; Interactive learning environments; Learning technology; Automated assessment; Mobile technology.

1. INTRODUCTION

Computer Science instructors have been exploiting learning technology such as Algorithm Visualization (AV) [1] for many years to explain hard-to-understand algorithms to the learners through simulations and animations. While many studies report its effectiveness in enhancing learner perception, Naps et al. [2] argued that these visualizations have little educational value unless it supports active engagement of a learner. Their *Engagement Taxonomy* [2] suggested six forms of students' engagement with AVs: no-viewing, viewing, responding, changing, constructing, and presenting. Additionally, the study concluded that "the higher the students' engagement with AVs, the better the educational benefits". Since then, many additional studies confirmed this hypothesis [3-5] and some [4-5] even redefined the original taxonomy to make it more contemporary and to better capture the expectations of learners and educators utilizing the AV systems. A recent study [5] argues that the abovementioned "Constructing" level, where students simulate the steps of an algorithm by manipulating AV interface, should be expanded to support automated assessment of construction exercises. Malmi et al. [6] strengthen the argument by stating that exercises may only help to reach learning goals if student's performance can be monitored and feedback is given immediately. However recent studies [3,5] reveal that there exist a few AV systems that support construction exercises, and among them, only two studies [5,7] support automated assessment. To date, in terms of engaging learners in technology-enhanced interactive constructing activities with automated feedback, both the number of efforts and our knowledge of its effectiveness are limited.

In an effort to explore the feasibility of developing an integrated learning environment that supports automated and immediate assessment of construction exercises, we developed Mobile Response System (MRS) [8-10]. MRS is a software environment that facilitates in-class construction exercises and their real-time assessment utilizing mobile devices. We argue that as mobile devices are becoming more pervasive and are used profusely by students, it will be intuitive for them to perform in-class construction exercises in their preferred devices. MRS is therefore designed as client server software that allows the instructor to dynamically prompt the students with carefully designed construction exercise apps, synchronized with the lecture material, in their mobile devices. Learners are able to actively interact with the exercise while recognizing the effect of their interactions visually at different stages of the algorithm and send their solutions back to the server computer. MRS then facilitates grading of the exercises automatically by comparing the learner made sequence of interactions with the correct sequence of interactions. Since the delivery, time keeping and grading of the exercise apps are automated in this technology-enhanced learning environment, instructor could use them more frequently in the classroom and learners could actively participate in more exercises where they can receive feedback. MRS also supports in-class anonymous communication, where students can send anonymous question/feedback to the instructor and additionally can vote on existing pool of questions that instructor may choose to review and answer at the end of the class. Another important aspect of MRS is the integration of instant data visualization that analyzes vast amount of student performance, interactions and mobile device usage data (i.e. button clicks, time spent, navigation behavior etc.) in order to summarize and interpret student performance, mental model, and attitudes during problem-solving.

The detailed design, implementation and deployment of the MRS system are out of scope of this paper. Interested readers are advised to explore MRS website [8] for downloadable modules and publications [9-10] to learn more. This paper explores MRS's support for engaging students via construction exercises and the

SIGCSE '17, March 08-11, 2017, Seattle, WA, USA
© 2017 ACM. ISBN 978-1-4503-4698-6/17/03…$15.00
DOI: http://dx.doi.org/10.1145/3017680.3017793

way it is utilized to create a technology-rich interactive classroom for students learning Algorithms. In this paper, we utilize case study, a step-by-step visualization of a construction exercise app, to explain how technology is leveraged to provide a richer way for learners to interact with a problem, and how instructor can acquire real-time evidence of student comprehension of covered lecture material. The paper also investigates whether the use of proposed visualization for construction tasks augmented with features such as automated assessment, anonymous question asking, and statistical data visualization yield better educational outcomes in a classroom. The rest of the paper is organized as follows: section two briefly describes the MRS system and construction exercise apps, section three presents case-study of one such visualization app, section four discusses the results of our investigation and section five discusses related research works.

2. MRS SYSTEM AND APPS

In MRS, instructor computer runs the server component of the software, which hosts questions, manages users, and maintains communication and synchronization. The client component executes in learner's mobile device, which allows her to login to the system and facilitates construction tasks. The server has been developed in Java and the client has been developed in Android. By exploiting MRS, instructor can import the construction exercises and broadcast them to learners' client devices. When the MRS client receives a new exercise from the server, the corresponding app that renders the given exercise is executed and allows learner interaction with the app. The app then captures learner interaction at every stage (screen) and sends it back to the server for grading. When the server receives answers back from all the clients, it uses runtime reflection to find the corresponding server-side grading component of the particular exercise and loads and executes the corresponding method in that component dynamically in order to grade learner submissions.

The most important aspect of the MRS software is the facilitation of construction tasks, where learners require to directly work on a visual representation of a problem and to develop the answer following a set of steps guided by a particular algorithm or process. In each step, learners make key choices (for example clicking the table or array indices for selection or swap, selecting from a drop-down menu, selecting a tree node or an edge of a graph etc.) that will impact their next step of interaction. While designing MRS system and relevant exercises, our goal was two-fold. Firstly, we wanted the exercises to be highly engaging and interactive to enforce deeper student learning. Secondly, we want the process of administering exercises as much effortless as possible for the instructors. These goals lead to the following design decisions:

- While performing these multi-step exercises, learners are able to go back and forth (by utilizing "Back" and "Next" button) and to compare the impact of their different choices.
- The exercises are designed as dynamic entities to support greater diversity and to allow learners to practice with different variations of the same problem. The exercises are therefore parameterized, where parameters can be populated with either randomly generated values or instructor generated values to create different instances of a problem. An exercise definition is stored as an XML file and contains exercise parameters such as problem components, time to answer, special instructions etc., which are used to render the corresponding app in the client device. The file additionally contains different rubric and grading parameters such as correct answers, grading weights, etc. to support server-side automated grading. Therefore creating a new exercise instance (with input data and grading rubrics) does not involve changing any programming code.
- Each exercise has a rubric associated with it, which grades partially correct answers as well to gauge learner's problem solving skills and cognitive models. Checkpoints are utilized in the client devices to monitor students' interaction throughout the exercise. The system then formulates a sequence of interactions as student submission and compares them with model sequence on the server side to determine how much of the exercise was answered correctly. In some instances, students may figure out the correct final answer of a problem by trying out all possibilities and without following all intermediate algorithmic steps correctly (e.g. finding the shortest distance between two nodes in a small/medium sized graph). The abovementioned grading process will enforce students to correctly follow the steps while simulating an algorithm.
- Students' device usage can also be tracked by setting the correct parameters in the XML file such as how often they utilize the "Back" button, how much time they spend in individual steps of a problem, how often they navigate to other apps (i.e. browsers, Tweeter or Facebook) etc.

3. CASE STUDY: PRIM'S MST

In computer science, Prim's algorithm is a greedy algorithm that finds a minimum spanning tree (MST) for a weighted undirected graph. Prim's algorithm builds the tree up by adding vertices one at a time to the current tree. The algorithm make it necessary to provide each vertex not in the current tree with the information about the shortest path connecting the vertex to a tree vertex and this information can change every time a new vertex is added to the tree. The concept of Prim's algorithm is therefore difficult for learners to learn and visualize. Usually learners trace pseudocode and draw diagrams with pen to understand how the algorithm works for a particular graph. In this case-study, we describe a construction exercise app that helps learners clarify their concept of Prim's algorithm and assess their own understanding via visual presentation, active interaction and hands-on nature of the approach. Few researches [11-13] developed Java-based interactive prim's exercise where given a graph, learners require to consecutively select (click) the next edge to be included in the MST following prim's algorithm. As "shortest edge" changes with evolving MST, we argue that, this prescribed way of interaction is neither adequate for clarifying confusions nor provides any visual cue for problem-solving.

In this proposed Prim's MST app, along with the graph, we also provide a table as part of the visualization where learners can keep track of the shortest distance to a non-tree vertex (by updating its parent and distance) everytime the MST evolves. Therefore, the learner interaction in this app is attained by clicking a graph node, manipulating drop-down menu items in a table, and by clicking navigation buttons (i.e Next, Back, Help, Submit Answer etc.). In step one (screen #1), learner selects the starting node of the MST as instructed by the instructor and click the "Next" button. Figure 1(a) shows the app scenario when a learner clicks node A (starting node) in the graph and is about to press "Next" to see the next interactive screen. It should be noted that when a node is included to MST, the corresponding node in the graph and the entry in the table are marked as color "Green". This is to signify that the table entry for the newly included node (i.e. parent and distance) can no longer be modified unless student utilizes "Back" button.

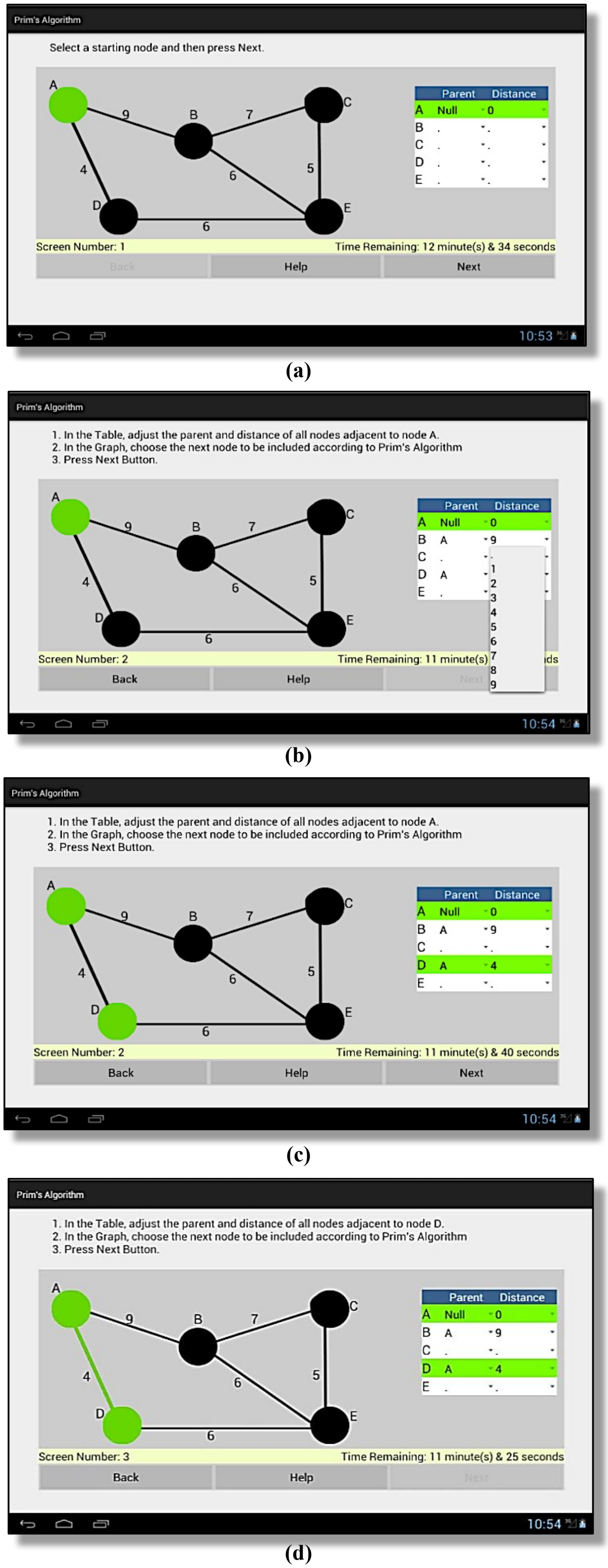
In each following step (screen), a learner must execute three tasks in sequence (see the app instructions in Figure 1(b)). Firstly, she needs to work on the table to adjust parents and distances of non-tree vertices that are adjacent to the recently selected tree vertex. Figure 1(b) shows the app screen where a learner is modifying node entries in the table that are adjacent to node 'A'. In this particular instance, she has completed modifying node 'B' (parent: 'A', distance:9) and is about to modify the distance of node 'D' by utilizing drop-down menu. Based on the information in the distance column of the table, the learner then applies prim's greedy strategy to find out the next node to be included to the MST and click that node in the graph. Figure 1(c) shows the situation when the learner decides to include node 'D' next, correspondingly clicks that node and is about to press "Next" to see the next interactive screen. Once "Next" button is pressed, the app automatically adds the appropriate edge to the MST based on the student provided parent information of the recently chosen node in the table (Figure 1(d)). The learner will repeat the same procedure until all graph nodes become part of the MST.

At any step, learner can click "Back" button to undo the previous selection and repeat that particular step. Therefore, it is possible for a learner to identify a mistake at the very last step, yet start from a clean slate again by utilizing "Back" button multiple times. Pressing the "Help" button opens the prim's algorithm in a separate window for learners to utilize during problem solving. The app captures learner activities at each step (screen) by storing the intermediate tables and graphs. The "Submit Answer" button becomes active only when the MST is complete (Figure 2) and submits learner interactions at each step of the app to the server for automated grading. The grading component of each app contain rubrics that can be customized to assess partially correct learner answers based on assigned weights. Each learner receives an email immediately containing the correct answer (i.e. the graph and the table) and a capture of the final screen of her submission.

MRS server also provides instant data visualization and analytic tools to share a wealth of information about the recently taken app exercise in the classroom. Figure 3(a) captures the "Grade Summary" screen that reflects the overall class performance of the exercise and various grade related statistics generated automatically and instantly for the instructor to share with the students. The "Time Taken" tab will allow the class to visualize timing data to identify degree of difficulty that majority learners faces while taking the quiz. The "Back Presses" tab shows "Back" button usage and "Navigation" tab shows how often learner navigates away to other apps during problem solving. Immediate availability of this assessment and tracking information allows instructor to gain better understanding of a learner learning of a concept and allows learners to assess her own progress toward the concept and to compare it with the rest of the class. Instructor can also share the correct answer with the class by utilizing the "Correct Answer" tab which shows the final graph, table and correct order of chosen nodes (Figure 3(b)).

4. EVALUATION

This section investigates whether the use of proposed visualization for construction tasks augmented with automated assessment, anonymous question asking and statistical data visualization features yield better educational outcomes in a classroom. In our formative assessment, we measure students' knowledge acquisition immediately after each treatment in the classroom. Whereas, our summative assessment investigates course failing rate and learners' satisfaction at the end of a course.



Prim's Algorithm

Select a starting node and then press Next.

Parent	Distance
A Null	0
B .	.
C .	.
D .	.
E .	.

Screen Number: 1 Time Remaining: 12 minute(s) & 34 seconds

(a)

Prim's Algorithm

1. In the Table, adjust the parent and distance of all nodes adjacent to node A.
2. In the Graph, choose the next node to be included according to Prim's Algorithm
3. Press Next Button.

Parent	Distance
A Null	0
B A	9
C .	.
D A	.
E .	.

Screen Number: 2 Time Remaining: 11 minute(s) ds

(b)

Prim's Algorithm

1. In the Table, adjust the parent and distance of all nodes adjacent to node A.
2. In the Graph, choose the next node to be included according to Prim's Algorithm
3. Press Next Button.

Parent	Distance
A Null	0
B A	9
C .	.
D A	4
E .	.

Screen Number: 2 Time Remaining: 11 minute(s) & 40 seconds

(c)

Prim's Algorithm

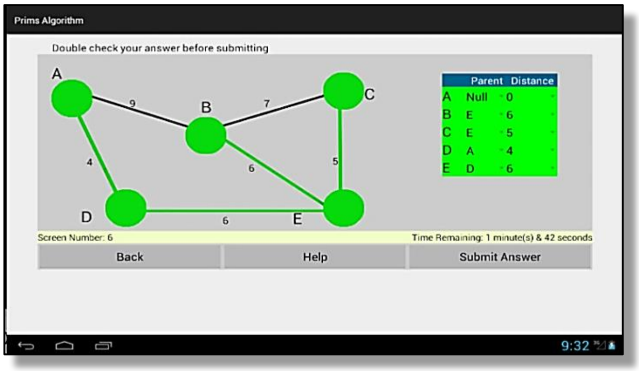
1. In the Table, adjust the parent and distance of all nodes adjacent to node D.
2. In the Graph, choose the next node to be included according to Prim's Algorithm
3. Press Next Button.

Parent	Distance
A Null	0
B A	9
C .	.
D A	4
E .	.

Screen Number: 3 Time Remaining: 11 minute(s) & 25 seconds

(d)

Figure 1. Prim's MST app Interactions



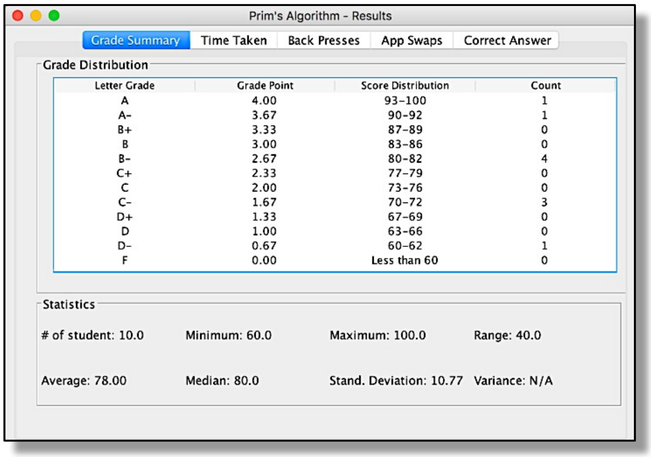
Double check your answer before submitting

	Parent	Distance
A	Null	0
B	E	6
C	E	5
D	A	4
E	D	6

Screen Number: 6

Time Remaining: 1 minute(s) & 42 seconds

Figure 2. Submit Answer is available as the MST is complete.



Grade Distribution

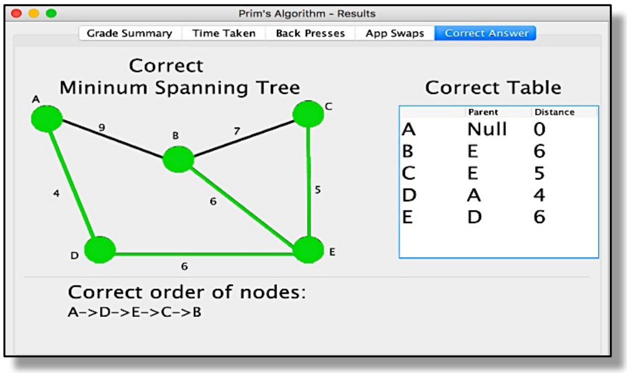
Letter Grade	Grade Point	Score Distribution	Count
A	4.00	93-100	1
A-	3.67	90-92	1
B+	3.33	87-89	0
B	3.00	83-86	0
B-	2.67	80-82	4
C+	2.33	77-79	0
C	2.00	73-76	0
C-	1.67	70-72	3
D+	1.33	67-69	0
D	1.00	63-66	0
D-	0.67	60-62	1
F	0.00	Less than 60	0

Statistics

# of student: 10.0 Minimum: 60.0 Maximum: 100.0 Range: 40.0

Average: 78.00 Median: 80.0 Stand. Deviation: 10.77 Variance: N/A

Figure 3(a). Grade Data Visualization



Correct
Mininum Spanning Tree

Correct Table

	Parent	Distance
A	Null	0
B	E	6
C	E	5
D	A	4
E	D	6

Correct order of nodes:
A->D->E->C->B

Figure 3(b). Correct Answer Visualization

4.1 Experimental Settings and Method

The MRS software and associated constructing exercise apps were deployed in an junior level "Analysis of Algorithms" course during the semesters of Spring 2015 (S15) and Spring 2016 (S16). During S15 semester, the study was conducted in a pilot fashion and because of the space limitation its formative results are not included in this paper. Table 1 shows some course and participant related information. During S16, three constructing exercise apps such as Dijkstra's shortest path (DSP), Prim's Minimum Spanning tree (MST) and Selection sort (SS)) were deployed in the classrooms, whereas in S15, SS and Bubble sort (BS) apps were utilized. The measurement instruments utilized in this evaluation are student performance data and the data collected from an anonymous student experience survey (IRB approved) conducted at the end of the semester.

As a methodology, we describe the sequence of events that occur in a typical class where instructor taught a topic (e.g. Prim's MST) and offerred construction exercise apps afterwards for students to be engaged with. During this deployment, in an 75 minutes long class, the instructor spent the first 30 minutes to explain Prim's algorithm and solving example exercises. It should be noted that students were familiar with greedy strategy and the use of MRS system and apps at that point, as previous classes were used to explain them. Then in the next 10 minutes, students applied their understanding of Prim's algorithm by interacting with the mobile app and solve a construction exercise (Attempt #1) which is very similar to the already covered examples in terms of degree of difficulty. As all students received instant email with their answers and correct answers after completing Attempt # 1, instructor then spent the next 5 minutes pointing out the mistakes and clarifying the concepts more. The following 10 minutes were spent by the students solving an exercise (Attempt #2) that is of higher order of difficulty than the examples and exercises covered previously. The next 10 minutes were spent by the instructor to reinforce the topic and correct student misconceptions (if any). The last 10 minutes of the class were spent by the students to perform a difficult exercise (Attempt #3) that requires critical thinking and involves unfolding scenarios that were not explicitly taught before.

4.2 Results and Discussions

As shown in Table 1, multiple problems based on the same concept (app) were offered until students show mastery of the concept and as a result students have more chances to practice the concepts and improve their comprehension. Also the class enjoyed significant anonymous interactions through question asking and voting as reflected in Table 1. Figure 4 shows the average grades that students of S16 semester achieved in their different attempts for learning DSP, MST and SS algorithms. It is clearly evident from this sequence of performance data that most students solved the first exercise successfully as it was very similar to the examples that were previously demonstrated in the class, however struggled for the second attempt as that was of higher order of difficulty, and finally did well for the last attempt once instructor was able to spend additional times correcting misconceptions and enhancing understanding. The automated assessment of exercises and class performance visualization data make it possible for the instructor to offer more hand-on-activities with different degree of difficulties to the students during the class time and utilize the extra time on correcting common misconceptions as identified by immdiate assesment. These results therefore validate the importance of performing multiple construction exercises with immediate feedback in order for students to gain proficiency on a concept.

Table 1. Course Related Data

	S15	S16
Number of Students	18	19
Female/Male	44%/56%	42%/58%
Number of apps used	2	3
Number of graded exercises using MRS	4	10
Number of Questions/feedbacks	Feature was not used	9
Number of votes on question/feedback	Feature was not used	16

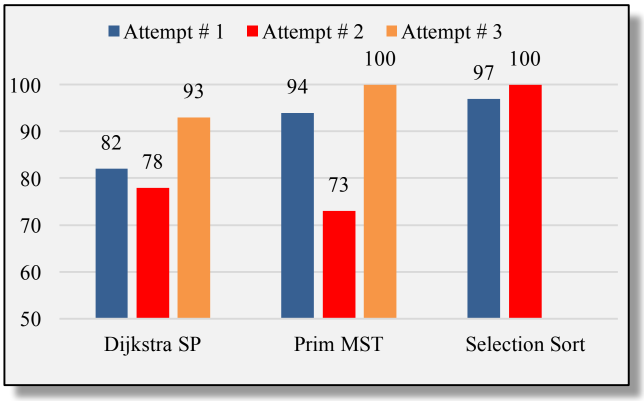


Figure 4. Student performance data for construction exercises

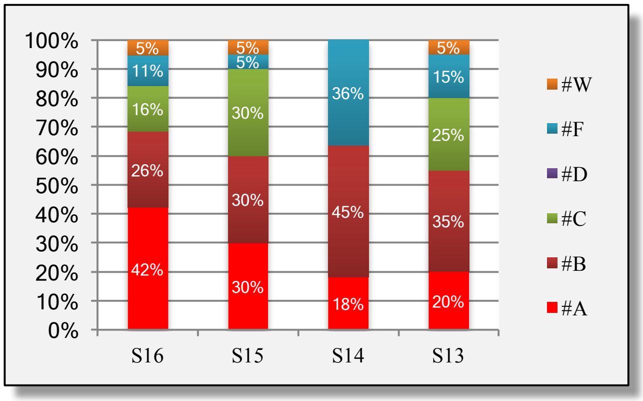


Figure 5. Overall course grade data

Table 2. Student Experience Survey Data

Question	SA	A	D	SD
Q1	75%	25%		
Q2	81%	19%		
Q3	88%	12%		
Q4	31%	53%	6%	
Q5	56%	44%		
Q6	75%	25%		
Q7			25%	75%
Q8	6%	6%	31%	57%

Q1. MRS is helpful in understanding Selection Sort, Prim's MST and Dijkstra's Shortest Path Algorithms.
Q2. MRS helps in visualizing the steps in the above algorithms.
Q3. Seeing my grades immediately after taking the App Exercises is really helpful.
Q4. The Selection Sort, Prim's MST Apps are intuitive and easy to use.
Q5. I feel competent and confident to take quizzes in MRS than in a pen-and paper-based setting.
Q6. I am enjoying this experience.
Q7. Using MRS did not improve my understanding of Selection Sort, Prim's MST and Dijkstra's Shortest Path Algorithms.
Q8. Learning to use MRS software and related Apps is substantial additional work beyond the normal course work.

Figure 5 shows summative results of overall course grades that students achieved in the course offerings for intervened S16 and S15 semesters and non-intervened S14 (N:11) and S13 (N:20) semesters. During the intervened semesters, learners utilized MRS environment and exercise apps to practice and solve exercises based on some core concepts, where the same exercises were offered in the classroom in a pen- and-paper-based setting during the non-intervened semesters. Same instructor taught all four offerings and the course setting and teaching practices have been consistently maintained both before and during the MRS interventions. Since the prerequisites (with a minimum grade of C) courses did not change between these offerings, we expect the student skill levels and background to remain fairly consistent across the different offerings of the same course.

Careful analysis of Figure 5 reveals that, collective number of students drop the course out (W grade) or did not take the final exam and receive a F grade reduced considerably when MRS is introduced. On average, 26% of student received either W or F grades in non-treated semesters and only 13% of students received the same during the treated semesters. Study [2] argues that drop-out and failure rate directly relate to learners motivation for stydying the topic and their attitude toward pedagogical methodology and tools used in the course. The performance data in Figure 5 therefore shows improved motivation and positive attitude toward the subject when MRS and construction AVs are utilized. This particular experiment also allows us to compare the educational benefit of using "constructing" enagement level (as in S15 and S16) with the "no-viewing" engagement level (as in S14 and S13) while teaching an Algorithm course. It is interesting to note that, although the number of students receiving an A grade improves from 19% (without MRS) to 36% (with MRS), the number of students receiving a grade B or higher remains nearly same (59% without MRS ad 64% with MRS). This observation indicates that MRS and construction exercise apps were most beneficial for a griup of students, who otherwise would receive grade B, however with MRS intervention they were able to enhance their understanding and achieved grade A. Hence our study confirms the superiority of "constructing" type engagement over "no-viewing" engagement.

Learners' satisfaction was measured with an anonymous experience questionnaire about the educational use of MRS and construction exercise apps. The students gave their opinion on eight statements using a Likert scale of four values such as Strongly Agree (SA), Agree (A), Disagree (D), and Strongly Disagree (SD). Table 2 summarizes the opinion question and results of the survey. It should be noted that for ensuring that students are actually looking at the content of the survey, not just "clicking through", the survey was designed to contain both positive and negative questions. As reflected in Table 2, the survey results represent a strong student approval of the proposed approach. Careful observation however reveals two interesting points. Firstly, a small group of students (6%) did not think that the apps are intuitive and easy to use. Although efforts were taken to familiarize the students with the app interface and interactions through tutorial and animations before the actual graded exercises, some students were observed to struggle with interactions while performing the exercises and this result supports that observation. Secondly, 12% students felt that learning MRS and apps is substantial overhead. These findings trigger the necessity of conducting proper usability studies for MRS software in order for it to be more intuitive and acceptable to the students. Student experience survey also contained a comment section where students provide feedback and suggestions based on their experiences. We created a bigram word cloud to visualize students' responses (Figure 6). The word cloud shows that students identified the most useful features of MRS (e.g. "immediate grading", "improved understanding" etc.) and used positive terms to describe their experience.

5. RELATED WORKS

Few researches leverage technology while supporting in-class interactive exercises and integrated assessment. TRAKLA2 [7] is

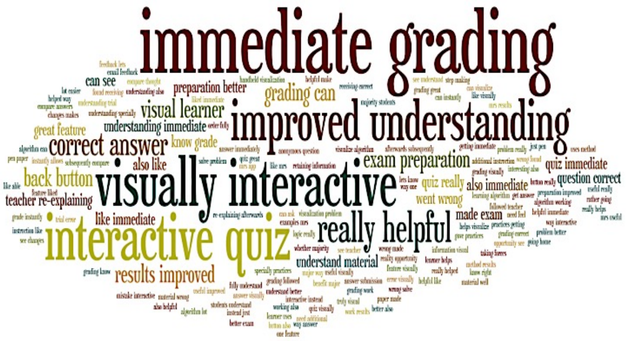


Figure 6. Word cloud based on student experience survey

the pioneering learning environment that supports algorithm simulation exercises, automatic assessment and visual feedback. JHAVÉ [11] is a java application that renders algorithm visualizations and allows learner explorations by pop-up questions. VisuAlgo [12] is an online collection of AVs that generates questions for the learners to answer based on the content. OpenDSA [13] and related JSAV [5] supports algorithm visualization and proficiency exercises. However all of them support content delivery via web browser and therefore utilized technologies such as HTML, Java, JavaSript etc. but none to our knowledge do so via native mobile app like MRS. The novelty of MRS in the context of active learning can be perceived by its support for the following:

- Multi-step construction apps with automated feedback.
- Dynamic/parameterized exercises.
- Storage of learner's interactions at every step of the exercise.
- Step-wise partial grading.
- Learner grade and attitude visualization.

6. CONCLUSION

The research presented in this paper implements an active classroom to teach Algorithms to CS students by exploiting Mobile Response System (MRS) and constructing exercise apps. Through MRS, instructor can exploit students' preferred devices to offer them multiple construction activities and can adapt her teaching based on the immediate and automated assessment, rather than waiting for all submissions to be graded manually. Our formative assessment data indicates that students' increased engagement with the mobile-friendly, interactive and instant-feedback nature of MRS allow them to comprehend the subject material better. The summative assessment shows significant reduction in course failing and drop-outs when students are intervened with MRS and constructing AVs through increasing difficulties and immediate feedback. The overall student performance data for four course offerings also confirms the necessity of including most engaging practices such as constructing when teaching Algorithms. MRS and constructing apps are found to be most effective for a group of students who otherwise would receive grade B, instead with MRS they ended up receiving grade A. Since MRS provided students the chance to correct misconception instantly and practice increasingly difficult concepts during the class, improved student learning and satisfaction are reported in the experience survey. However, some students complain about the extra work needed for learning to use MRS and apps.

In future, we would like to emphasize more on usability study of the MRS software in order for it to become intuitive and effortless. We are also receiving a wealth of student performance, interaction and mobile device usage data during each offering and in future we would like to explore this big dataset to investigate some interesting patterns such as "the more a student utilize the back button, the better is her grade" or "Timing data can be a precursor of student misconception or limited understanding of a topic" etc.

7. ACKNOWLEDGMENTS

This research was supported by National Science Foundation Award #1332531.

8. REFERENCES

1. C. A. Shaffer, M. Cooper, & S. H. Edwards, "Algorithm Visualization: A Report on the State of the Field", In Proceedings of the 38th SIGCSE technical symposium on Computer science education, 2007.
2. T. L. Naps , G. Rößling , V. Almstrum , W. Dann , R. Fleischer , C. Hundhausen et.al., "Exploring the role of visualization and engagement in computer science education", ACM SIGCSE Bulletin, v.35 n.2, 2003.
3. J. Urquiza-Fuentes , J. Ángel Velázquez-Iturbide, "A Survey of Successful Evaluations of Program Visualization and Algorithm Animation Systems", ACM Transactions on Computing Education (TOCE), v.9 n.2, p.1-21, 2009.
4. N. Myller, R. Bednarik, E. Sutinen, and M. Ben-Ari, "Extending the engagement taxonomy: Software visualization and collaborative learning", ACM Transactions on Computing Education (TOCE), v.9 n.1, 2009.
5. V. Karavirta, C. A. Shaffer, "Creating Engaging Online Learning Material with the JSAV JavaScript Algorithm Visualization Library," in Learning Technologies, IEEE Transactions on, no.99, p.1-1, 2015.
6. L. Malmi , A. Korhonen , R. Saikkonen, "Experiences in automatic assessment on mass courses and issues for designing virtual courses", Proceedings of the 7th annual conference on Innovation and technology in computer science education, 2002.
7. L. Malmi, V. Karavirta, A. Korhonen, J. Nikander, O. Seppä̈lä̈, and P. Silvasti. "Visual algorithm simulation exercise system with automatic assessment: TRAKLA2", Informatics in Education, v.3 n.2 p.267–288, 2004.
8. Mobile Response System (MRS), http://compsci.wssu.edu/MRS/, 2016.
9. Fuad M. M., Deb D., James E. and Gloster C., "Using Interactive Exercise in Mobile Devices to Support Evidence-based Teaching and Learning", 21st ACM Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE), Arequipa, Peru, p. 17-22, 2016.
10. Fuad M. M., Deb. D. and Etim J., "An Evidence Based Learning and Teaching Strategy for Computer Science Classrooms and its Extension into a Mobile Classroom Response System", Proceedings of the 14th IEEE International Conference on Advanced Learning Technologies (ICALT), IEEE Press, Athens, Greece, p. 149-153, 2014.
11. T. L. Naps, "Jhavé: Supporting algorithm visualization", Computer Graphics and Applications, IEEE, v.25 n.5 p.49-55, 2005.
12. S. Halim, Z. C. Koh, V. B. H. Loh, and F. Halim, "Learning algorithms with unified and interactive web-based visualization," Olympiads in Informatics, v.6, p.53–68, 2012.
13. E. Fouh, D.A. Breakiron, S. Hamouda, M.F. Farghally, & C.A. Shaffer, "Exploring students learning behavior with an interactive eTextbook in Computer Science Courses", Computers in Human Behavior n. 41, p. 478-485, 2014.