Reflections on advocating visualization as part of an introductory programming

Correlie, M. Pretorius
Tshwane University of Technology
South Africa

Albertus, A.K. Buitendag
Tshwane University of Technology
South Africa

Darelle, van Greunen
Nelson Mandela Metropolitan University
South Africa

Abstract
Today, more than ever, innovative approaches are needed to encourage and catalyze the teaching and learning activities of students. This is becoming increasingly true in the South African tertiary education domain. This research presents the results obtained from a questionnaire based on the utilization of videos, as part of a blended learning approach, to support the notion of the application of visualization as part of the problem solving processes for an introductory programming course. The questionnaire focused on obtaining data relating to the composition and packaging of the content as well as student's thoughts and ideas with regard to the presentation thereof and how it aided their studies. The data collected are analyzed and used to improve the development and teaching strategies of future subject packaging and presentation. The questionnaire contained both qualitative and quantitative data that was analyzed and presented. Some of the key findings and observations include the ability to re-watch and review the content until the subject content is perceived to be mastered, it also supports the fact that content should be presented within the reference context of the students and that humor could have an effect on the understanding and comprehension of topics. Other findings include statistical data that is elaborated further on in the study.

Keywords
Algorithm construction, Basic programming principles, Cause and effect teaching, Problem solving, Visualization, Teaching innovation

Introduction and background

Introductory ICT related courses often include a module or subject relating to computer programming and problem solving. Presenting and managing a subject such as an introduction to computer programming has its own challenges that require unique

Steyn, J., Van Belle, J.P. (Eds.). 2015. Beyond development. Time for a new ICT4D paradigm?
Proceedings of the 9th IDIA conference, IDIA2015, Nungwi, Zanzibar: 134-147
ISBN: 978-0-620-68395-1.

interventions and approaches, especially when it relates to novice programmers.

Countries and institutions worldwide face numerous challenges relating to the design of Information and Communication Technology (ICT) and Computer Science (CS) curricula, in all sectors including primary, secondary and tertiary institutions of learning (Brown, Sentance, Crick, and Humphreys, 2014; Rubio, Romero-Zaliz, Mañoso and de Madrid, 2015).

Some of these challenges include motivating students to pursue studies in the ICT and CS disciplines cf. (Giannakos, 2014) as well as obstacles in the development of courses, materials and assessments for introducing students to the core concepts of programming and problem solving, cf. (Grover, Pea and Cooper, 2014). In countries such as the US and in continents like Europe there has been a decline in the numbers of students enrolling for ICT related courses. (Goode and Margolis, 2011; Sadler, Sonnert, Hazari, and Tai, 2012)

Grunfeld, (2011:10) explain that “ICT for Development (ICT4D) research projects are normally heavily oriented towards the application of information systems in developing countries, among individuals, in communities, government and businesses”. A significant factor intrinsically predominant as a requirement is the level of education of the users of the system. The subsequent development and maintenance thereof by developers are influenced. The need to present and develop course content to kindle and promote students to pursue ICT related studies, and to succeed in their initial perusal of their studies are therefore imperative. Various factors influence student dropout rates, in the first year of study. The retention of students is important as can also be seen from other studies, cf. (Aguiar, Chawla, Brockman, Ambrose, Goodrich, 2014; Morgan, Sheard, Butler, Weerasinghe, 2015)

Introducing basic programming principles at the University of Technology (UT) creates challenges and makes the subject one of high risk. Therefore lecturers and subject heads are continuously attempting and introducing innovative new teaching ideas with the aim of improving the success rates of the subject. The improvement of the success rate is threefold, firstly to attempt to increase the average pass rate for the subject, secondly to have a high impact on the cognitive problem solving skills development of the students, and thirdly to aim and improve the retention of first year students. This activity also forms part of a managerial directive to improve the teaching and learning practices of lecturers and to improve the presentation of content. Students enrolled at UT are required to complete a lecturer and course assessment instrument, for each registered subject bi-annually.

As part this study an additional instrument was developed. This instrument is used as a guideline for improvement, and feedback obtained is analyzed to better package course content and to highlight some areas of improvement.

One aspect that is evident from experience is that students often lack comprehension relating to the question posed and problems to be solved. This is also notable in the way in which students approach answers in assessments and exercises. Bachu and Bernard (2014) support this notion by explaining that some of the challenges students face as part of an introductory programing course include program comprehension, program

generation and programming fluency. It is further emphasized that: “student’s lack of problem solving skills remains one of the biggest challenges”.

Blackwell, Whitley, Good and Petre (2001) explains that part of the challenges novice programmers experience reside in the fact that the cognitive processes involve to “separate processing of syntactic and semantic information, the collection of expert knowledge into chunks, the structuring of regularly-used information into schemas, and the solution of design problems in terms of previously acquired and frequently modified plans.” Ahoniemi and Lahtinen (2007) also supports this notion and presented that novice programmers often face the challenge of developing concrete models and comprehending problems of abstract concepts.

The pilot study

One of the predominant issues identified by the researchers is the fact that many students struggle to comprehend not only the scenario but also the problems posed as part of questions. Some of these problems were highlighted in answers presented by students as part of previous assessment attempts. Comprehension with regard to language also plays a role. During the first part of 2014 the researchers identified the need to develop a set of videos with the aim to present programming and problem solving related content in a visual manner.

Primary research objective

The primary objective of the study and the development of the subsequent lessons were to determine the effects that visualization and the promotion thereof as part of an introductory problem solving and programming curriculum could have on the cognitive development of the students and success rate of the applicable subject.

Visualization as a teaching tool

The concept of visualization and the application thereof in the presentation of programming related subjects attract a lot of research cf. (Ahoniemi and Lahtinen, 2007; Shaffer,Cooper, Alon, Akbar, Stewart,Ponce and Edwards, 2010; Bachu and Bernard, 2014; Glassman, Scott, Singh, Guo, and Miller, 2015)

Moreno, Myller, Sutinen and Ben-Ari (2004) are of the opinion that visualization, based on several studies has proved to be a successful tool to help in the development of students programing skills as long as the students are cognitively involved and as long as the visualization are targeted for the particular user population.

Hundhausen, Douglas, and Stasko, (2002) defines Algorithm visualization (AV) as various methods that is applied to graphically represent how computer programs act, and is often developed with the aim to promote the understanding of how algorithms work.

Shaffer et al. (2010), points out that: “a growing body of evidence indicates that certain uses of AV do have a measurable impact on student learning. The most important factor appears to be engagement of the students’ attention”.

Design of the video subject matter

As part of this study, a total of nine video lessons were developed, based on a real life scenario that students could relate to. These video lessons incorporated generic principles relating to the application of visualization techniques. Actors were used to act out these

scenarios before the algorithm to solve the problem was explained. Errors commonly made by students were also illustrated where after the actors would act out the effect of the mistakes on the end users in order to emphasize and illuminate the importance of valid and logically correct calculations steps and statements.

The lessons focused on the application and promotion of visualization by including and incorporating various forms of visualization such as:

- Algorithm visualization (AV) as defined by Hundhausen, Douglas, and Stasko, (2002)
- Program visualizations (PV) in a lesser extent as described by Urquiza-Fuentes and Velázquez-Iturbide, (2009) as "The visualization of actual program code or data structures, low-level abstraction, in either static or dynamic form." This aspect was partially incorporated from the view of the algorithms construction and subsequent execution.
- Scenario visualization (SV) defined by the researchers as the mental models to recognize the scenario and problem domains presented as well as the cause and effect of certain steps placed as part of the logical flow of a proposed solution. SV could also include aspects of visualization that allows the viewer's thereof to: "explore the variation in solutions at scale". (Glassman et al., 2015). For the researchers SV also encapsulates the idea of active thinking along all the processes of algorithm design. This includes visualization relating to the analysis of the problem statement, input processes, processing logic and output

Lesson design

Lessons are scenario and theme-based and were developed to include aspects of all three elements of visualization as listed above.

Figure 1 presents the general construction and flow for each lesson.

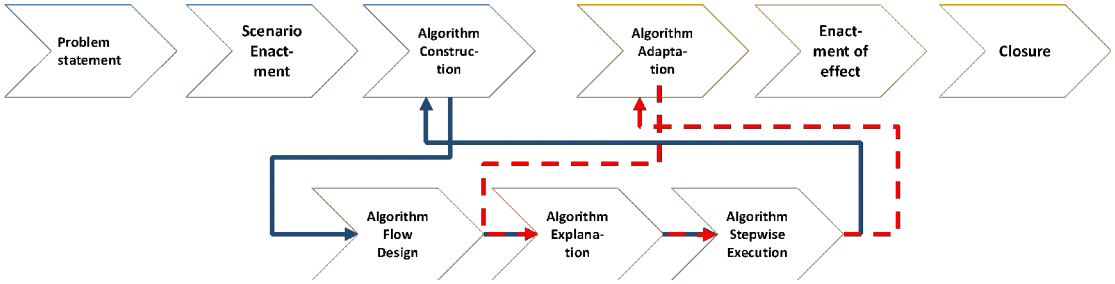

Figure 1. Generic lesson flow and construction

Each lesson started off with a real world problem statement that resembles a typical problem statement from an assessment or assignment. The actors performed a script relating to the problem statement that is based on a bigger scenario. In this performance, the actors placed emphasis on the requirements of the problem statement posed as well as the ultimate objectives, such as a calculation relating to income, ticket sales, expenses, etc. Next the algorithm was constructed in a visual way with emphasis being placed on the logical thoughts and thinking relating to the inputs and outputs required, based on the enactment. The construction of the algorithm was presented visually, and the execution thereof also visually explained. The construction presented the logical flow and design as well as the motivation for presenting the steps in the selected manner. The stepwise execution was supported with animations relating to the calculations. This was done in order to show how input is obtained and processed in order to display the required output.

Alternate ways of approaching a problem was also highlighted in lessons were applicable. To support the teaching and learning process and to kindle the notion of cause and effect, the algorithm was then adapted to include common types of mistakes students typically make as experienced by the researchers. These mistakes were then supported and illustrated by animation, as well as an enactment by the actors to show the effect that small changes for example could have on the outcome of an operation. One important aspect conveyed in this approach was to present the notion of, think before you construct, and to thoroughly test with sample data.

The predominant pedagogical motivation for the development of the lessons was to:

- promote the understanding of the basic concepts of programming,
- present the concepts as part of a scenario with smaller events based on situations and settings which are familiar to the majority of students,
- impose the idea of scenario visualization as part of the planning process in the development of an algorithm for later translation into a high level programming language and
- enforce the use of visualization to make abstract programming concepts more concrete.

The primary driver for the motivation above is to help students to overcome their mental barrier regarding the initial analysis of a problem, and to present the relationship between the real life and the problem posed.

Table 1 presents the primary content and objectives of each of the lessons and also presents the scenario on which the lesson was based.

Lesson Nr	Primary objectives and content covered	Lesson scenario	Additional content highlights
1	Explain basic programming concepts, relating to variables, operators and operations	“Making Millions” is a reality show in which 3 learners (Mpho, Thabo and Precious) made it through to the finals where they will each receive R10 000 in order to start their own small business for a 3 week period. The presenter of the show, Maxine, will join them during this period on their journey to see what challenges they experience and to determine who the winner will be. The main theme is explained and the characters introduced. Lesson objectives are explained based on the main theme and the characters.	A song is used to explain and help the students to remember the precedence of operators. Attributes and behavior of the actors are used to explain and support the attributes of variables and basic calculations.
2	A problem is solved explaining the sequence control structure.	Thabo decided to use a part of his R10 000 to rent a mini-bus for the three week period in order to utilize it as a taxi. As the cost to rent the mini-bus is calculated based on the days rented as well as the kilometers driven, he estimates the kilometers per day in order to calculate the amount that he should pay at the end of the 3-week period.	When Thabo receives the mini-bus, he experience problems with a flat tyre. The concept of an algorithm and steps executed in sequence are explained by means of the steps Thabo retrieved from the internet to replace the flat tyre. To explain the execution of the algorithm, memory positions of variables are visually depicted and a

			computer screen is also visually illustrated at strategic points to link variables with input and output when entering input and displaying output.
3	A second problem is solved explaining the sequence control structure. All calculations are executed in sequence.	Mpho decides to start a spaza stall (small shop) selling apples and chips. She wants to sell the apples in small packets of 5 each, but would like to retain at least 10, should anybody want to buy a single apple. The program logic assists with the packaging and calculation of expenses and income.	Integer division and modulus arithmetic are explained in detail and in a visual manner. Students can see how calculations can be simplified by means of intermediate variables.
4	A problem is solved explaining the selection control structure. It also contains statements in sequence.	Precious retained her work in a Shebeen (pub) while planning to make her debut in the entertainment industry and to host a show. The algorithm calculates her wage for her current shift in the Shebeen.	Various ways of writing the same nested if-statement to solve this problem are explained and illustrated. It is pointed out how easy it is to write a nested if-statement that looks perfectly valid, yet will execute in an incorrect manner due to invalid conditions.
5	A second problem is solved explaining the selection control structure	Mpho decided to expand her spaza stall by baking and selling muffins. This problem explains that when buying flour, how she would determine which quantity would be the most economical to buy.	Once again various ways of solving the problem are provided when a nested if statement is rewritten as a compound if-statement to obtain the same results.
6	A third problem is solved explaining the selection control structure	Thabo decided to manage his small business in an honest manner and to take out insurance for the 3-week period. This video deals with the calculation of the insurance premium.	A complex problem statement with many conditions and corresponding if-statements is analyzed and explained.
7	A problem is solved explaining the iteration control structure. It focuses on the for-loop, yet contains selection statements as well.	Mpho buys 6 types of ingredients for her baking, some which are on special and implies discounts.	Emphasis is placed on when calculations should be done inside the loop and when it should be done outside the loop.
8	A second problem is solved explaining the iteration control structure. The structure is rewritten to explain the do-while pre-test loop and loop-until post-test loop.	Precious hired a hall to host her first performance. This problem focuses on the ticket sales for the show. The audience could qualify for one of 2 types of discounts – either the student/pensioner discount or the group discount for groups of 10 or more.	The complexity of compound conditions in loops is explained. Entertainment is included in the video as a motivation factor. While explaining her plans to Maxine, Precious performs a song to the customers in the Shebeen as a preview to attract more customers to the show.
9	A third problem is solved explaining loops – in this case nested loops.	Thabo illustrated his daily routes and how he accumulates income per day, per week and eventually for the complete period of three weeks.	The complicated nested loops (one per day, one per week and another for the complete period) are explained in a step by step manner, highlighting aspects to be aware of when writing nested loops for accumulation processes. The series of videos is concluded in a humoristic way with a few “behind the scenes” shots.

Table 1. Synopsis of the lessons developed

Figure 2 presents some snapshots from scenes of lesson 5 (including the problem statement) as well as some slides used to support the notion and implementation of AV for illustration in conveying the content. The sample lesson content and flow presented in figure 2 conform to the construction of a lesson as depicted in figure 1. Each video lesson starts off with a problem statement, the enacting of the problem statement followed by a visual explanation of the algorithm logic to solve this problem. Some mistakes commonly made by students are then included in the algorithm to illustrate the effect it would have on the output. This is followed by a last performance by the actors to illustrate the effect that incorrect statements could have on end users of a computer program or system.

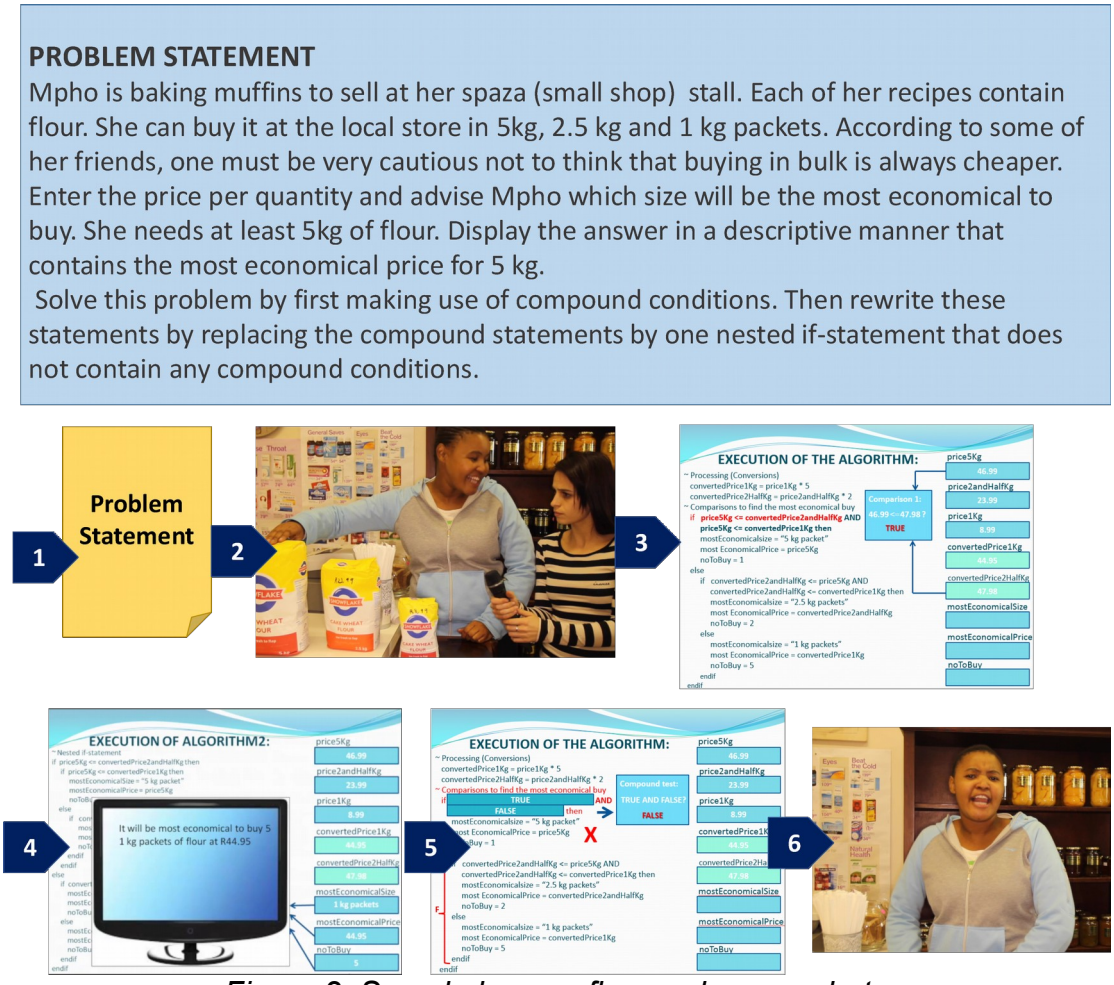

Figure 2. Sample lesson, flow and screenshots

Methodology

The predominant method of enquiry used as part of this study was that of a survey which was presented in the form of a questionnaire. The design of the questionnaire included aspects relating to the student's experiences towards their own learning. Aspects such as, how the use of the lessons could enhance their understanding of the subject content and related application thereof were also included.

The initial plan of the application of the video lessons was to include the use thereof as part of the normal instruction given to students, in the second semester of 2014. Due to various factors the use of the videos was postponed. The first use of the videos was therefore promoted to students on a voluntary basis during semester 1 of 2015 as part of an introductory programming course for which 732 students registered at Soshanguve South campus. The instrument and videos were also made available for other distance campuses, however for the purposes of this report only the Soshanguve student's responses were included, due to the proximity of the researchers and the students. This is the campus where the researchers reside. The students were informed of the videos, its intended purpose and use. Students were also encouraged to watch the videos as part of their own development processes and, reinforcement of the subject content and for enrichment purposes. Presenting this option to the students resulted in the automatic creation of three groups of students, which is: 1) those who decided not to watch the videos, 2) those who watched some or part of the videos and, 3) those who watched the complete set of nine videos. Using this strategy fundamentally created two groups the target group (those students who watched all of the videos) and the reference group (those students who did not watch all of the videos). The students who did not watch any of the videos were discarded for the purpose of this research report. Some of the students who participated repeated the subject. Table 2, presented next, indicates the target group and reference group in terms of participants and how many of these students repeated the subject. The subsequent discussions relate only to the target group as being deemed relevant, due to the fact and perspective that these students watched all the videos, regardless of them repeating the subject or not. It is important to note that, the target group comprised of students who decided on their own to watch all of the videos, indicating some form of dedication and willingness to go to extra mile.

	Target group	Reference group	TOTAL
First time students	98	195	293
Repeater students	18	25	43
	116	220	336

Table 2. Target and reference group numbers

Design of the instrument (Questionnaire)

Various types of questions where included and utilised as part of the instruments design which included typical yes or no answers, open ended questions, Likert scale questions using a scale of 1 to 10 and short multiple response type questions. The design of the instrument was done in order to capture and obtain data relating to the following aspects i.e.:

1) generic use questions, such as the approach towards the utilization of the videos by the student
2) the way in which the students could relate to the scenarios and characters in the videos
3) approach towards the presentation of the lesson objectives and the construction of the algorithm in terms of inputs, outputs and the steps involved
4) effects of the promotion of visualization in terms of the algorithm and subsequent execution
5) cause and effect principle towards the notion of accuracy when planning and designing an algorithm
6) effect of humor to present a mental mode towards the learning process

7) support towards the mastering of the main subject outcomes
8) general comments and perceived educational value

Each of the eight aspects above and the data obtained were used to aid in deriving the answer the primary objective of this research which aims to determine the effects that the promotion of the visualization practice could have on the success rate of an introductory programming course.

Results and analysis

As this study predominantly focuses on user experience, the data collected of the only the target group was analyzed and the results of each of the aspects presented towards the primary research objective are presented next. The data from the reference group was not used as part of this study as comments relating to the videos and their perceived benefits were analyzed and not the inherent results or performance.

Reflections on the generic use and approach towards the utilization of the videos

Of the 18 repeater students in the target group, 17 students (94.5%) indicated that they watched 1 or more videos more than once and 7 students (38.8%) indicated that they watched all 9 videos more than once. The students indicated that the predominant reason for re-watching the videos was to re-enforce understanding. Of the 98 new students in the target group, 87 students (88.8%) indicated that they watched 1 or more videos more than once and 39 students (39.8%) watched all 9 videos more than once. It was also evident that students deemed the educational value of the videos high as 11 repeaters (61%) indicated that they always took notes and 5 repeaters (28%) sometimes took notes. For the new students, the figures are as follows: 50 students (51%) always took notes and 42 students (43%) sometimes took notes. Many students (93%) also indicated that they sometimes or even often had to rewind to watch a certain part of the video again. This could be for re-enforcement but could also indicate that the pace could have been too quick.

Ability to relate to the scenarios and characters in the videos

On a scale from 1 to 10 where 1 being “strongly disagree” and 10 being “strongly agree”, most students in the target group indicated that they could relate to the characters and scenarios used in the videos. The average for repeater students was 7.3 and for new students it was 7.45.

Perception on lesson presentation regarding achievement of lesson objectives and input, output and steps involved

Most students perceived these aspects of the video lessons as positive as can be seen in the following table that contains the average of their answers on a scale from 1 to 10:

Steyn, J., Van Belle, J.P. (Eds.). 2015. Beyond development. Time for a new ICT4D paradigm?
Proceedings of the 9th IDIA conference, IDIA2015, Nungwi, Zanzibar: 134-147
ISBN: 978-0-620-68395-1.

	Average: repeater students	Average: new students
Importance of planning	7.12	7.75
Different ways to solve a problem	7.3	7.87
Effective ways to solve a problem	7.5	8.1
Test run with sample values	6.6	7.85
Use of IPO-chart during planning	8.14	7.75
Mastering basic concepts first	8.4	8.04

Table 3. Student perceptions and towards the videos

78% of repeater students and 71% of new students also indicated that the videos assisted them to understand IPO-charts, as part of a Yes/No question.

Effects of the promotion of visualization in terms of the algorithm and subsequent execution

The target group students were positive about the effect of visualization to explain how the algorithm executes and variables are populated with values. (On a scale from 1 to 10, new students scored it an average of 7.55 and repeater students 6.4)
They also indicated that after the watching the videos they tend to visualize the problem before they start to solve it (average of repeater students: 7.4 and that of new students: 7.46) and they tend to visualize the problem in terms of the input and output required (average of repeater students: INPUT: 8.53 OUTPUT: 8.31 and new students: INPUT: 7.83 OUTPUT: 7.7)

How students perceived the cause and effect principle towards the notion of accuracy when planning and designing an algorithm

The researchers are pleased to report that this aspect of the questionnaire yielded the most positive results. It seems as if students could really identify with the characters when they were very upset with incorrect computer results. Two questions were posed on this topic and it was interesting to note that in the repeater group of the target group no student rated the output less than 5 for the first question and less than 6 for the second question. The averages for these two questions are as follows:

	Average: repeater students	Average: new students
Small changes in the algorithm could have an effect the output and on the end users	8.27	8.06
Students realized the importance of accurate and reliable output to end users	9.13	8.2

Table 4. Student perceptions towards promoting the cause and effect notion

Effect of humor to present a mental mode towards the learning process

Surprisingly, this was the question that yielded the lowest average when students were asked to rate it on a scale from 1 to 10. The average for the 18 repeater students was 5.73 and the average for the 98 new students only 5.67. However, from comments such as "*Fun and informative*", "*PERFECT . . . FUN*" and "*They are easier to understand than sitting in front of book. They also provide us with a fun way of solving problems*" it may be considered that some students did enjoy the humoristic aspects of the videos.

Support towards mastering of the main subject outcomes

All target group students were positive (with an average more than 70% per question) that the videos assisted in mastering IPO charts, variables, priorities in execution of operators, analysis of the problem statement and the understanding of selection statements and the for-loops. They were less positive about the mastering of pre-test and post-test do-loops. This could be due to the fact that these topics were covered only towards the end of the course and that there was not enough time for this. However, it could also be that there were not enough examples on these two loops in the videos. However, there were comments where students expressed their gratitude that the videos helped them to distinguish between pre-test and post-test loops such as "*The videos really helped me to understand the do while and do until better because I did not really understood [sic] in class*".

General comments and perceived educational value

When asked to provide comments on the videos in general and on the video lesson that they found most interesting, 5 (28%) of the repeater students and 39 (40%) of the new students did not provide a comment. The number of positive repeater student comments was 12 (67%) whilst 48 (49%) new students perceived the videos as positive through their comments. However, if these figures are expressed according to the number of comment responses received, it increases to 92% positive repeater comments and 81% positive comments from new students. The results indicate the need for further study, investigation and provide optimal motivation for further enhancements.

The following comments were received from the 2 top achievers in the target group that adds substantial value to the notion.

- "*I found lesson 8 the most interesting because it covers almost all we've learned in DSO. It helped a lot!! For example, to know when to initialize inside and outside a loop. I also liked Precious's song ;) It is really good!! Since am not a fan of books it helped me in understanding the theory part of C++*"
- "*Video lesson 1 was my favorite. This video lesson introduces the basic programming principles. The beginning of programming is in the design. This is where students should start, with the introduction. These videos helped me understand programming principles and I recommend other visual learners like me to participate in educational video lessons.*"

Discussion

From the reflections and review of each of the eight aspects as discussed above, towards attaining the main research objective it is evident that that the promotion of visualization principles including scenario visualization (SV) were beneficial. The utilization of the videos was predominantly reflected and perceived as positive and contributed towards the final success rate of the subject.

Three other Likert scale questions that supports the results, presented the following average scores. The questions:

- "Videos like these will encourage students to become enthusiastic about programming and problem solving", resulted in a score of 7.36 for the repeater students and 7.85 for the new students.
- "I would love to have videos like these in follow up subjects", which refer to other programming subjects presented results of 8.25 from the repeater students and 7.57 from the first time students.
- "I will recommend these videos to other students", calculated to an average score of 8.2 for repeater students and 7.76 for the first time students.

One important fundamental aspect toward the use of the videos by the students, relate to initial drive and motivation of the students to watch the videos i.e. Target group. It could be presented that the primary objective was not just only to get a better understanding of the subject content, but also ultimately to pass the subject.

Some generic comments presented by students included toward the following suggestions for improvement by:

- incorporating small exercises and quizzes as part of each video, and presenting a discussion on each.
- presenting additional scenarios in the videos which could then be enacted learners in class or as part of tutorial group sessions.
- breaking down the videos into smaller segments with additional practice opportunities.

Conclusion

The initial motivation for this study related to the promotion and inclusion of visualization strategies as part of a first-year introductory programming course. Visualization and the application of the cause and effect strategy, in the planning process, contribute positively towards the comprehension of the problem posed. Another aspect includes the practice as a tool to help students to better understand abstract concepts. It is also evident, that the implementation of visualization contributed positively to the motivation of students, based on their overall comments. More than one student also mentioned the value of being able to watch the videos in their own time, until they perceive content to be mastered. The results obtained presented very positive feedback and provide the opportunity for further studies and enhancements. Concrete suggestions for the better design of such videos were derived and these aspects could easily be implemented in the development of further lessons and AV tools to better package the introductory content. It is evident from the

study that the positive feedback received and the subsequent results obtained by the students, necessitates the continuous advocating of the visualization practice, not only in the first year of study but also in latter related subjects. Further studies could also focus more on other aspects relating to the retention of students, the enhancement of subject presentation and success rates. Stimulating and developing students towards the better application of problem solving skills, could only be beneficial towards the ICT4D objectives overall.

References

Aguiar, E., Chawla, N.V., Brockman, J., Ambrose, G.A., Goodrich, V., 2014. Engagement vs performance: using electronic portfolios to predict first semester engineering student retention, In: Proceedings of the Fourth International Conference on Learning Analytics and Knowledge. ACM, pp. 103–112.

Bachu, E., Bernard, M., 2014. Visualizing Problem Solving in a Strategy Game for Teaching Programming, In: *Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering* (FECS). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), p. 1.

Blackwell, A.F., Whitley, K.N., Good, J., Petre, M., 2001. Cognitive factors in programming with diagrams. Artificial Intelligence Review 15, 95–114.

Brown, N.C.C, Sentance, C., Crick, T., & Humphreys.S., 2014. Restart: The Resurgence of Computer Science in UK Schools. Trans. Comput. Educ. 14, 2, Article 9 (June 2014), 22 pages. DOI=10.1145/2602484 http://doi.acm.org/10.1145/2602484

Glassman, E.L., Scott, J., Singh, R., Guo, P.J., Miller, R.C., 2015. OverCode: Visualizing variation in student solutions to programming problems at scale. ACM Transactions on Computer-Human Interaction (TOCHI) 22, 7.

Goode, J., Margolis, J., 2011. Exploring Computer Science: A Case Study of School Reform. ACM Transactions on Computing Education 11, 1–16. http://doi:10.1145/1993069.1993076

Grover, S., Pea, R., Cooper, S., 2014. Promoting active learning & leveraging dashboards for curriculum assessment in an OpenEdX introductory CS course for middle school. ACM Press, pp. 205–206. http://doi:10.1145/2556325.2567883

Grunfeld, H., 2011. The contribution of information and communication technologies for development (ICT4D) projects to capabilities, empowerment and sustainability: a case study of iREACH in Cambodia. Victoria University.

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

Moreno, A., Myller, N., Sutinen, E., Ben-Ari, M., 2004. Visualizing programs with Jeliot 3, in: Proceedings of the Working Conference on Advanced Visual Interfaces. ACM, pp. 373–376.

Morgan, M., Sheard, J., Butler, M., Weerasinghe, K.F.S.A., 2015. Teaching in First-Year ICT Education in Australia: Research and Practice, In: Proceedings of the 17th Australasian Computing Education Conference (ACE 2015). p. 30.

Rubio, M.A., Romero-Zaliz, R., Mañoso, C., de Madrid, A.P., 2015. Closing the gender gap in an introductory programming course. Computers & Education 82, 409–420. http://doi:10.1016/j.compedu.2014.12.003

Sadler, P.M., Sonnert, G., Hazari, Z., Tai, R., 2012. Stability and volatility of STEM career interest in high school: A gender study. Science Education 96, 411–427. http://doi:10.1002/sce.21007

Shaffer, C., 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 10, 1–22. http://doi:10.1145/1821996.1821997

Sengupta, P., Kinnebrew, J.S., Basu, S., Biswas, G., Clark, D., 2013. Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. Education and Information Technologies 18, 351–380. http://doi:10.1007/s10639-012-9240-x

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 9, 1–21. http://doi:10.1145/1538234.1538236