Teaching Programming using Arduino-based Educational Frameworks

Tumelo Makgaka University of Cape Town
Rondebosch, Cape Town
mkgtum005@myuct.ac.za

ABSTRACT

Computer Science is a course with a consistently high failure and dropout rate across multiple universities. Lecturers often struggle to get students to relate with the abstract concepts taught in the discipline and this causes students to have a level of intimidation and confusion towards the subject, which decreases their enthusiasm to continue further. To curb the declining interest in computer science from students, lecturers need to constantly attempt to find ways to make the course content interesting and engaging for students. This project aims to assess whether or not the use Arduino based educational frameworks can improve student interest and enthusiasm in programming assignments and computer science courses. This research will assist lecturers and universities in making informed decisions about whether or not creating programming assignments with physical components is a worthwile option to pursue in the attempt to improve student interest. The benefit is assessed by having students complete two assignments where they complete the code for a networked alarm system. Once they complete the assignments, the time they take to complete the assignments and their enthusiasm levels for the assignments are recorded. The first assignment is one which contains physical Arduino components and the second assignment is a simulation based assignment written in java. In both assignments, students were given an incomplete program with explanatory diagrams, and were expected to fill in the missing code. The experiment was initially intended to survey first year computer science students but was ultimately conducted with 6 postgraduate computer science students from the university of Cape Town. Once they completed the assignments their time of completion was recorded and they provided expert user feedback. Overall, the physical assignment created greater levels of enthusiasm towards the task than the simulation assignment. The physical and simulation assignments mantained similar levels of task completion times among students.

DOI: 10.1145/1235

1. INTRODUCTION

Programming knowledge is fast becoming a necessity in the modern world, knowledge that is necessary for not only computer scientists, but also designers and engineers looking to create interactive prototypes[1]. Computer Science and IT related professions are also considered scarce skills in South Africa[6][14]. Despite this need for people to become literate in programming, many find introductory programming courses difficult[7][10][11], resulting in high failure rates[2][3] and declining enrollments[13][16][12][2].Students also lack motivation to complete the course[10] and find the content and assignments boring[10].

It is imperative that a new approach to introductory Computer Science education is found, particularly one that manages to interest and engage students better.The Arduino is an open source hardware project that has created a series of microcontrollers designed for users to build hardware devices and interactive prototypes. The Arduino board is coded in the Arduino Language via an open source IDE.

A method that has currently been employed is the use of microcontrollers, specifically the low-cost, open-source Arduino boards[15][9][4][5][1], both have been successful in stimulating interest in students and motivating them to complete their assignments and investigate further topics in Computer Science.

This project will attempt to introduce students to basic Computer Science topics by giving them an assignment where they complete the code for a networked alarm system using the arduino IDE and a java simulation with the identical physical components to assess the effect that a physical relation between their code and the real world has in engaging students and provide them with additional motivation to not only complete the assignment.

This research is necessary since it will assist in understanding the relationship between the presence of physically implemented concepts and student interest in programming education, along with the effect of physically implemented concepts on the time students take to complete an assignment as opposed to concepts implemented through a simulation. It also aims to assess whether student preferences differ in each assignment. Since creating new assignments and implementing new teaching methods requires a lot of time and effort from lecturers who may not have previous experience with micro controllers and it may be expensive for the university to buy a micro controller for every student in an introductory programming course, this research will assist lecturers to better assess whether or not implementing such assignments is worth the effort.

2. PROBLEM STATEMENT

Through our research, we aim to understand whether or not the introduction of a physical aspect to Computer Science assignments using Arduinos could engage students interest and motivation to complete the assignment and even learn more. We seek to answer the following questions with our research:

- What impact does the assignment completion method, be it physical or simulation have on student interest in programming education ?
- What impact does the assignment completion method, be it physical or simulation have on the time students take to complete an assignment?

These questions are important as their answers could assist in the implementation of Educational Frameworks and assignments in the future, so that students can become further engaged and interested in Computer Science. The answers are also useful in informing the university in whether or not purchasing micro controllers and training tutors and lecturers in how to use micro controllers is a worthwhile expense, whether the benefits outweigh the costs.

3. RELATED WORK

There have been previous experiments done to assess the impact of tangible programming on student interest and enthusiasm [8][17].In these experiments, students created programs using physical components which represented various programming constructs, such as if statements, loops and variables[comment on results of experminets and how they differ to what I want]. In an experiment conducted by Rubio[15], Arduinos were used to reinforce programming concepts for students. Arrays were taught using an array of sounds emitted from a loudspeaker, loops were taught using a motor and conditional statements were taught using LED's. The effectiveness of the teaching methodology was assessed through using a class's previous score as a control variable and measuring the performance of the class after the arduinos were used against the control. The results showed an increase in student performance in the Arduino class and generally positive feedback from students on the inclusion of the Arduinos in their curriculum. The accuracy of the experiment results however may be confounded by the fact that the control class and experiment class were conducted a year apart from each other, which makes room for the testing conditions to vary in a way that is not anticipated.

4. SYSTEM DESIGN

The alarm system functions by having the user arm it and detecting motion in a confined space, then when motion is sensed the buzzer is rang and the user disarms it by entering the password using the remote. The alarm system has the following functionality:

- Arm alarm
- Disarm alarm
- Change alarm pin

The simulation software was created by making use of java swing GUI elements, namely JButtons and a text field. A grid of buttons was made to represent the remote, the text field represented the LCD screen and the motion sensor image represents the PIR motion sensor. The motion sensing aspect is simulated by moving the mouse over the image.

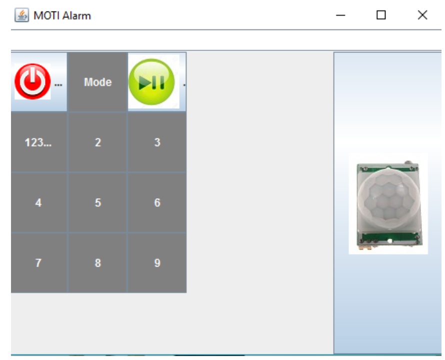

Figure 1: Simulation GUI

The physical system used in the testing sessions was designed by assembling an alarm system using Arduino components. The components that were used for the alarm are an Arduino uno micro controller, PIR motion sensor, infrared sensor, remote, lcd screen and a pezio buzzer.

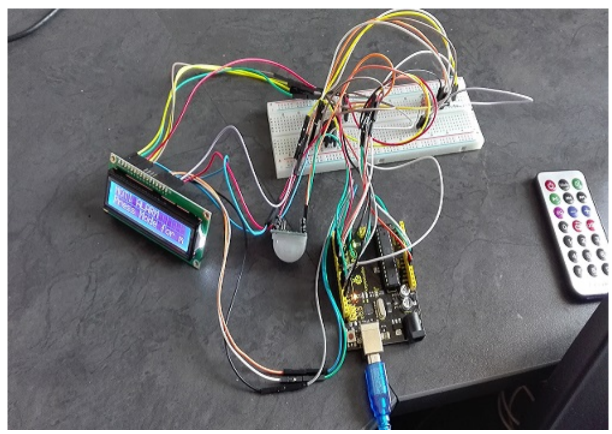

Figure 2: Physical system

5. EXPERIMENT DESIGN

The experiment contains a qualitative and quantitative aspect. The qualitative aspect of the experiment relates to the question of whether or not students will enjoy programming more if a physical component is added to it and the quantitative aspect of the experiment will asses which assignment students take a shorter time to complete. The experiment also aims to assess whether or not there is a relationship between the task time and the enthusiasm.

The experiment was designed by creating 2 identical assignments for students to complete. In the first assignment, students completed the skeleton code for the Arduino and in the second one, they completed the skeleton code for the java simulation. Students were expected to fill in the same method bodies in the assignments. Before they begun with the assignment, they were briefed about the methods they were expected to complete and what was expected to happen when the program executes. Once they acknowledged that they understood what they were meant to do, the timing of the assignment begun. To ensure that there were no carry over effects that could have hindered the results of the assignment, students alternated between starting with the simulation assignment and the physical assignment.

To ensure that students had an assignment that they could complete within the allocated time, the functionality of the original intended assignment was scaled down to only being able to arm and disarm an alarm.The dependent measures of the study are student enthusiasm and the time of assignment completion. The measures were collected through the questionnaire completed by students and through timing students while they completed the assignments.

5.1 Confounds

Prior to the experiment being conducted, there were confounds present that could affect the legitimacy of the test results. The confounds are as follows:

5.1.1 *Prior experience*

Students that have prior experience with micro controllers may be more comfortable with working with them in the assignment, and therefore could finish quicker than other students. This confound was solved through pre assembling the Arduino micro controller and pre coding lines of code related to the Arduino. An example of such a line of code is where the input of the PIR motion sensor is retrieved.

5.1.2 *Difference in appearance of systems*

The simulation system being created by using a Java GUI may add an additional confound through creating a bias towards the physical system, which could be seen as providing a more realistic system. Due to the absence of critical components in simulation softwares such as 123d and Fritzing, this confound could not be alleviated, and as thus could only be taken note of.

5.1.3 *Reduction in expected participants*

Another confound was through the number of student participants being reduced, which could prevent statistical deductions from being made. This confound was circumvented through having expert users test the system, namely honours and masters students.

5.1.4 *Students not finishing assignment*

This confound prevents an accurate analysis of results, since when students go above the allocated time, their time is simply recorded as recorded by the maximum time allowed and not the time that they actually would have taken. This confound was fixed by reducing the amount of tasks students need to fulfil in the assignment.

5.1.5 *Presentation order of assignments*

Another confound that could affect the study is the presentation order in which the assignments are given to students. To ensure that there were no carryover effects from students completing one assignment and going on to another, such as becoming better experienced at the assignment or getting tired, the order in which the options for the assignments was presented was counter balanced using a full counterbalancing method. For example, if student A did the simulation assignment first, then student B did the physical assignment first.

6. RESULTS

Due to the disruptions in the normal teaching schedules of the University, the intended plan of the experiment which involved surveying 20 first year computer science students could not be carried out. As a counter measure, the assignment was completed by expert users 6, namely honours and masters students from the university of Cape Town. The assignments were completed within 2 testing rounds, which occured within a space of 2 weeks.

Students were given 15 minutes to familiarize themselves with the different components and 30 minutes to complete each assignment. The analysis of the results was split between the time students took to complete the assignments and their enthusiasm for each assignment. All students managed to complete the assignment.

In the second round of testing the feedback that students gave about the assignment, which included comments about the lack of substantially descriptive comments in the code were solved.

6.1 Time taken to complete the assignment

The analysis of the results relating to the time students took to complete the assignment was conduced by performing a linear mixed model fit on the data. Before the linear mixed model can be performed, it is necessary to ensure that the data that will be tested meets the ANNOVA assumptions. The ANNOVA assumptions are met when:

- Each subject (participant) is independently sampled
- The data is normally distributed
- Has homogeneity among it's variance (Homoscedasticity)

Since the subjects were sampled independently, the remaining assumptions to satisfy are the normal distribution and homoscedasticity assumptions. A Kologmorov Smirnov test was done to ensure that the data follows a log normal distribution. The result of test was as follows

- $p < 0.8664$ - Simulation
- $p < 0.7749$ - Physical

A p value greater than 0.05 for both the Simulation and Physical informs us that the data does not have a significant departure from the log normal fit, and that it follows a log normal distribution.

Homoscedasticity was ensured by carrying out Levene's test on the data. The following p value was obtained as a result:

- $p < 0.53$

A p value greater than 0.05 for the test informs us that the homoscedasticity assumption is met.

To assess the impact that the method students complete an assignment in has on the time students take to complete an assignment, a linear mixed model was used.The obtained F values is as follows:

- Method $F_{1,11}$=0.0815

The significant F statistic for method ($F>0.05$) tells us that the method students complete the assignment in, be it the physical method or simulation method affects the time they take to complete the assignment .

6.2 Student enthusiasm

A recurring problem that students had with the simulation in the feedback of the first round of testing was that it had significantly more scaffolding than the physical assignment, which was as a result of creating the GUI for the assignment.They felt that the amount of code present in the program gave them a hard time understanding the program in its entirety.

7. FINDINGS

When completing the assignments, students had similar levels of performance in both assignments. The time they took when completing the physical and simulation assignments had a 0.23 second difference between the average times. This is likely due to the time students were given to familiarize themselves with the assignment and ask questions, which removed any ambiguity from the assignments. When the results were analyzed, the linear mixed model indicated that the method students use to complete an assignment has an impact on the time they take to complete the assignment, however the effect is evidently not strong. The effect may change if more participants are used and if first year as opposed to expert users are used as participants.

Students significantly preferred the physical assignment over the simulation. The difference in enthusiasm between the physical assignment and simulation was by over 30 percent. This may be as a result of the novelty of the deviation from normality of the assignment method, since computer science assignments generally consist of students fulfilling physically intangible tasks. The enthusiasm towards the physical assignment may also be influenced by the difference in appearance in the two assignments. Since the Java simulation was created using Java swing components, the implementation created aesthetic limitations to matching the physical system. The levels of enthusiasm for the assignment however did not seem to improve the time students took to complete the assignment, with only a less than second difference in the average task times of the physical assignment and simulation.

In the survey results, some students felt that they would have enjoyed the simulation assignment more if it was longer, and also if they were able to complete more tasks in the assignment. Other students however felt that they would only enjoy such an assignment if the micro controller comes pre assembled for them, since they viewed the connection as intimidating. All the surveyed students felt that the physical assignment they participated in made them more interested in learning more about micro controllers.

Students in the second round of testing on average completed the assignments quicker than in the first round of testing. They also felt more enthusiastic about the assignment in the second round of testing as opposed to the first round. This may be attributed to the students having a better idea of what the assignment entails, and familiarizing themselves with the arduino board. Students were also pleased with the solving of the problems they cited in the previous testing round, which included a comment about the lack of substantial explanation behind some of the skeleton code. An example of this is a grevience a student had with the Java skeleton code, where students were required to complete their code in a pre written thread, which was not done in the Arduino based program.

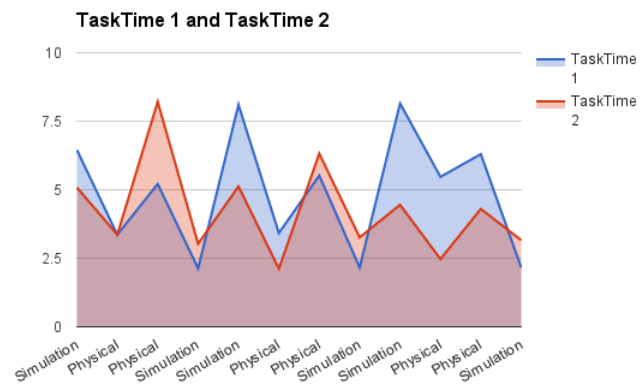

Figure 3: Test round 1 vs 2 task time

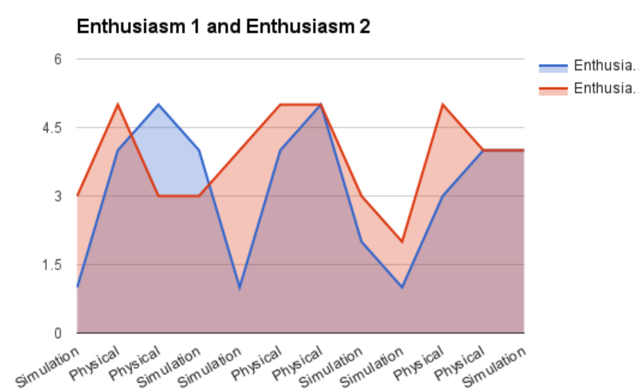

Figure 4: Test round 1 vs 2 enthusiasm

8. CONCLUSIONS

The results from the experiment show a positive outlook from the expert users with regards to the use of physical assignments. The use of physical components managed to improve their enthusiasm levels on the assignment, as opposed to the simulation assignment which is a popular method of assignment in computer science classes. The usefulness of physical assignments in programming education is however questionable, since the students performed similarly in both assignments. This leads to the decision of make use of physical components in programming assignments ultimately being an economic one. In a university where resources are scarce and there is a fixed budget for expenditure, simulation assignments may be seen as the best option since they are cost effective, but still manage to fulfill the task of assessing students.

In an institution that can afford to source the necessary physical components, the decision to use physical assignments in programming education can also be determined by the objective of the lecturer in the course they are teaching. If the objective is to improve student enthusiasm in programming and decrease the dropout rate, then it is worthwhile to create an assignment similar to the physical assignment created in the experiment. If the objective of the lecturer is only to improve student performance, a simulation assignment may be the better option, since it requires no additional cost to the university and is quicker and easier to implement, since it does not require the lecturer to learn about micro controllers and teach tutors about them.

The results however do not allow us to draw conclusions on effectiveness of physical assignments on student performance (time taken to complete the assignment), since all the participants had extensive prior knowledge in the programming constructs presented in the assignments. The low number of participants also presents a low confidence level in the accuracy of the results. A retrial of the experiment is necessary to allow us to draw a conclusion to whether or not the use of physical components in programming education significantly increases student performance and enthusiasm levels.

9. FUTURE WORK

9.1 System design

The system can be improved by implementing the simulation with a prototyping software such as 123d or Frtzing. This will provide a better comparison system, since the physical and simulation assignment systems will be identical, which will help in removing a bias towards the physical system which may be percieved as more realistically looking. The simulation software could not be used for this assignment due to missing components in the software that were part of the physical assignment. This problem can be circumvented by exluding the missing components in the physical assignment to ensure an identical system can be created using the simulation software.

9.2 Methodology

Since the survey was conducted using expert users, this is likely to have decreased the time students took to complete the assignment and may have a significant deviation from how long the intended participants (students) would have taken to complete it. The experiment can be improved by conducting it with more participants, particularly first year students. An increase in participants will also require the use of tutors, and to ensure that students all receive an equal level of assistance, the tutors all need attend a facilitated briefing on the assignment where they are expected to complete it. An additional text assignment can be added to the experiment to assess whether or not the inclusion of a visual component to assignments actually improve student enthusiasm and performance.

10. REFERENCES

[1] S. Alers and J. Hu. Admoveo: A robotic platform for teaching creative programming to designers. In *Learning by playing. Game-based education system design and development*, pages 410–421. Springer, 2009.

[2] J. D. Bayliss. Using games in introductory courses: Tips from the trenches. In *ACM SIGCSE Bulletin*, volume 41, pages 337–341. ACM, 2009.

[3] J. Bennedsen and M. E. Caspersen. Failure rates in introductory programming. *ACM SIGCSE Bulletin*, 39(2):32–36, 2007.

[4] L. Buechley and M. Eisenberg. The lilypad arduino: Toward wearable engineering for everyone. *Pervasive Computing, IEEE*, 7(2):12–15, 2008.

[5] L. Buechley, M. Eisenberg, J. Catchen, and A. Crockett. The lilypad arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 423–432. ACM, 2008.

[6] R. Daniels. Skills shortages in south africa: A literature review. *DPRU Working Paper*, (07/121), 2007.

[7] A. Gomes and A. J. Mendes. An environment to improve programming education. In *Proceedings of the 2007 International Conference on Computer Systems and Technologies*, page 88. ACM, 2007.

[8] M. S. Horn and R. J. K. Jacob. Designing tangible programming languages for classroom use. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*, TEI '07, pages 159–162, New York, NY, USA, 2007. ACM.

[9] P. Jamieson. Arduino for teaching embedded systems. are computer scientists and engineering educators missing the boat? *Proc. FECS*, pages 289–294, 2010.

[10] T. Jenkins. On the difficulty of learning to program. In *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*, volume 4, pages 53–58, 2002.

[11] M. Koorsse, C. B. Cilliers, and A. P. Calitz. Motivation and learning preferences of information technology learners in south african secondary schools. In *Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists*, pages 144–152. ACM, 2010.

[12] S. Kurkovsky. Engaging students through mobile game development. In *ACM SIGCSE Bulletin*, volume 41, pages 44–48. ACM, 2009.

[13] S. Leutenegger and J. Edgington. A games first approach to teaching introductory programming. In *ACM SIGCSE Bulletin*, volume 39, pages 115–118. ACM, 2007.

[14] B. E. Nzimande. List of occupations in high demand: 2014. *Department of Higher Education and Training*, 2014.

[15] M. A. Rubio, C. M. Hierro, and Á. Pablo. Using arduino to enhance computer programming courses in science and engineering. In *Proceedings of the EDULEARN13 Conference*, pages 5127–5133, 2013.

[16] K. Sung. Computer games and traditional cs courses. *Communications of the ACM*, 52(12):74–78, 2009.

[17] P. Wyeth and H. C. Purchase. Tangible programming elements for young children. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '02, pages 774–775, New York, NY, USA, 2002. ACM.