Programming education through a framework based on developing a tool on a micro controller

Tumelo Makgaka[1]

Abstract— **This paper reviews various litterature centered around programming education to assess the effectiveness of using micro controllers as an educational tool to teach programming. It initialy makes the assessment of what problems beginner programmers face, what learning methods exist and also assesses previous work done in teaching micro controllers or using them as an educational tool. The paper also assesses the practicality of the solution in a first year computer science class. Based on previous litterature, it is then established that creating an educational framework to teach programming using a micro controller would best be implemented in a series of class exercises done in groups that would require large amounts of support for students with a class on electronics, a sufficient amount of tutors to answer student questions,descriptive diagrams in exercises explaining how to accomplish tasks such as connecting perhipial devices to an arduino and scaffolding in the form of method headers in the programs students are meant to work on.**

I. INTRODUCTION

When students first learn how to program they are introduced to abstract concepts which may be difficult for them to understand. According to statistics from various universities across the world [1][2][3], student retention in computer science majors has constantly been on the decline, with less students registered for the major in first year graduating with it in their final year. The low retention of students in computer science majors is particularly prevalent with female students[4].These alarming statistics inform us that lecturers need to adopt a teaching style for computer science which will engage students more effectively.

The purpose of this review is to critique existing litterature on the education of programming, in order to understand how micro controllers can be used to enhance the learning experience in a programming class. We will assess litterature on the challenges beginner programmers face, the best teaching styles for teaching computer science and how micro controllers have been taught and used to enhance the computer science learning experience.

The review is divided into the following sections: (1) difficulties beginner programmers face,(2)how we can best curb the difficulties and (3) what previous work has been done in using micro controllers to teach programming.

II. WHY STUDENTS FIND PROGRAMMING DIFFICULT

In order to effectively create a framework for teaching programming, we first need to understand what hinders the success of students in doing well in programming. Once we have a clear understanding of what students find difficult, we can respond accordingly to adress the problems.

A constant theme in previous research articles shows that students generally feel they are unable to fully grasp programming concepts due to the abstract nature of the discipline. These are concepts where students feel that they can not receive high amounts of feedback on, which they can use to measure their progress. Students tend to understand concepts like syntax better due to the rapid feedback they are able to receive, which in turn allowed them to rectify their shortcomings[5].

The set of programming concepts that students recurringly had problems with were object oriented concepts[1][6]. Students felt as though object oriented concepts were abstract and required a different way of thinking from what they were used to. This is a key point to note due to the popularity of object oriented languages as the first language taught to students[6].

Over the years there has also been a decline of females registering to study computer science[4].Another problem that needs to be solved is the existing perception of Computer science [3]. Students were shown to avoid selecting CS as their major due to existing perceptions they had about the discipline, where they believed that all they'd be doing is sitting behind a desk programming. The perception that students have about the what happens in the discipline may also translate to their peception of their capabilities with regards to the discipline[7]

III. APPROACHES TO IMPROVING LEARNING EXPERIENCE IN PROGRAMMING

To establish which is the most appropriate approach for teaching programming, it's essential to understand how students learn. There are two main teaching methods which exist in Pedagogy(the practice of teaching), the constructivist and the objectivist apporach.

The constructivist approach assumes that students learn by assimilating different experiences they have had together, and the objectivist approach assumes that students learn by having a teacher teach them, essentially that a student is an empty vessel until a teacher fills the vessel up with knowledge[8].

Numerous studies [8] [9] [10] have been done to see which method of learning students prefer to learn programming and they have all pointed towards a constructivist approach. In order to effectively implement a constructivist approach to learning programming, we will need to identify teaching methods that are inline with the theory. Examples of appropriate teaching methods are having a flipped class room, where students do preparations at home before coming to

class, then once class starts they are asked questions. At the core of the constructivist approach is the idea of letting students be in control of deciding what they have learned, and then asking them questions about this [10]

Students feel that sample programs help, more physical approach, self study. With regards to practical assignments, sutdents also prefered that the assignments be broken up into smaller parts as opposed to having one large assignment[11]. This helps students to grasp the subject matter better since they have less of a chance of feeling overwhelmed by the content

IV. Use of micro controllers in teaching programming

Embedded systems are defined as "microprocessor based systems that are built to control a function or range of functions"[10]. Embedded systems have many applications today, from carrying out the operations in coffee vending machines such as inputting money to security alarm systems.Micro controllers form the focal point of embedded systems.The logic of the system is coded on to a micro controller and any other desired peripheral devices are attached to it. To create a suitable framework on using micro controllers to teach programming, we will look at previous applications of micro controller education as educational tools

Through their interaction with micro controllers, students can become more well rounded software developers through learning how to use low level languages such as c and assembly[13][14].The use of micro controllers in the education of programming brings the opportunity of being able to present programming in a different way by taking things from an on screen and visual perspective to a physical perspective[13]

It also brings the opportunity of being able to find a way to teach programming in a way which appeals to both genders[16].Having students work on creating things which appeal to both genders(i.e fashion through wearable technology) has been shown to get more females interested in computer science and also encourage students to transfer the skills they learn to other disciplines

Students were also shown to learn how to use micro controllers more effectively when they had representational diagrams guiding them on how to carry out various activities[14]. This assisted them in not getting carried away with the details of the actual micro controller, but rather focusing on solving the problem at hand

Using Arduino devices as the micro controllers of choice is advantageous due to its open source design. This means that components for the Arduino can be found easily and at a relatively low cost, there is a good online community present which can help students with problems they may encounter and there are various libraries available which can help students avoid reinventing the wheel[15]. The provision of libraries make the Arduino Microcontroller more user friendly and this trait makes it more inline with our aim of using micro controllers as an educational tool as opposed to attempting to teach students how to use micro controllers.

V. DISCUSSION

The previous litterature on approaches to improving learning experience also did not speak on whether the constructivist approach is appropriate in large classrooms. This is an important question since the approach is centered around students learning by developing their own experiences, which as a byproduct brings in many questions from students.If there are not enough tutors present or the tutors are not equipped enough to answer questions, this may present various problems

The litterature also did not discuss the impact of getting students to solve a problem that students can relate to.This question is important, since it's related to the subjectivity behind what people find as interesting.It informs us on whether students will find any problem presented to them that they need to address using micro controllers interesting, or whether the idea needs to have certain characteristics for them to find it interesting. It essentially lets us know if students are interested in the technology they will use to solve the problem, the problem itsself or a combination of the two

One key fact we need to be aware of when designing the framework is that Micro controllers as a module is an in depth subject which is typically taught in an embedded systems class. Many students find the micro controllers classes to be hard and abstract[17].In previous litterature, the students had prior experience in electronics [12] or electronics was added to the course material [15].

The amount of content students need to learn to be proficient at using micro controllers brings the potential for students to be overwhelmed by the content.This brings forth the requirement of planning exactly we want students to learn. The aim of the module will answer questions on how much support should be provided to students and how much scaffolding should be added to the programs they will create.

VI. CONCLUSIONS

In an entry level computer science course it will not be feasible to teach students how to use micro controllers in depth.We can use Arduinos as a novel way of teaching students how to develop programs through using lots of scaffolding in the course. This may entail creating skeleton programs for students with function headers and have them code the function bodies. This may also entail using diagrams to give step by step instructions on how to assemble an arduino and its peripheral devices together.

We will make use of a constructivist approach to teach this through presenting the content in the form of class exercises. Being mindful of the low tutor to student ratio, the solution will most likely best be implemented in student groups,with tutors to supervise each group during class exercises so that students can ask as many questions as they want and get rapid feedback on concepts they do not understand

Although through the course we will not be able to teach students on a wide range of applications of Micro controllers, the course will have a lasting impression since it will provide students with the entry knowledge they need to experiment

with micro controllers further and develop a greater interest for the discipline of computer science

REFERENCES

[1] Theresa Beaubouef and John Mason. 2005. Why the high attrition rate for computer science students: some thoughts and observations. SIGCSE Bull. 37, 2 (June 2005), 103-106.

[2] Pivi Kinnunen and Lauri Malmi. 2006. Why students drop out CS1 course?. In Proceedings of the second international workshop on Computing education research (ICER '06). ACM, New York, NY, USA, 97-108.

[3] Lori Carter. 2006. Why students with an apparent aptitude for computer science don't choose to major in computer science. In Proceedings of the 37th SIGCSE technical symposium on Computer science education (SIGCSE '06). ACM, New York, NY, USA, 27-31.

[4] J. McGrath Cohoon. 2001. Toward improving female retention in the computer science major. Commun. ACM 44, 5 (May 2001), 108-114.

[5] Gomes A, Mendes AJ. Learning to program-difficulties and solutions. InInternational Conference on Engineering EducationICEE 2007 Sep 3 (Vol. 2007).

[6] Butler, Matthew, and Michael Morgan. "Learning challenges faced by novice programming students studying high level and low feedback concepts." Proceedings ascilite Singapore (2007): 99-107.

[7] Susan Bergin and Ronan Reilly. 2005. Programming: factors that influence success. SIGCSE Bull. 37, 1 (February 2005), 411-415.

[8] Mordechai Ben-Ari. 1998. Constructivism in computer science education. In Proceedings of the twenty-ninth SIGCSE technical symposium on Computer science education (SIGCSE '98), Daniel Joyce and John Impagliazzo (Eds.). ACM, New York, NY, USA, 257-261.

[9] Essi Lahtinen, Kirsti Ala-Mutka, and Hannu-Matti Jrvinen. 2005. A study of the difficulties of novice programmers. SIGCSE Bull. 37, 3 (June 2005), 14-18.

[10] Psztor, A., Pap-Szigeti, R. and Trk, E.L., 2010. Effects of using model robots in the education of programming. Informatics in Education, 9(1), pp.133-140.

[11] Tuukka Ahoniemi, Essi Lahtinen, and Teemu Erkkola. 2007. Fighting the student dropout rate with an incremental programming assignment. In Proceedings of the Seventh Baltic Sea Conference on Computing Education Research - Volume 88 (Koli Calling '07), Vol. 88. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 163-166.

[12] Heath, Steve (2003). Embedded systems design. EDN series for design engineers (2 ed.). Newnes. p. 2. ISBN 978-0-7506-5546-0.

[13] Rubio, Miguel A., Carolina Manoso Hierro, and A. Pablo. "Using Arduino to Enhance Computer Programming Courses in Science and Engineering." Proceedings of the EDULEARN13 Conference. 2013.

[14] Bolanakis, D. E., E. Glavas, and G. A. Evangelakis. "An Integrated Microcontroller-based Tutoring System for a Computer Architecture Laboratory Course." International Journal of Engineering Education 23.4 (2007): 785.

[15] Zachariadou, K., K. Yiasemides, and N. Trougkakos. "A low-cost computer-controlled Arduino-based educational laboratory system for teaching the fundamentals of photovoltaic cells." European Journal of Physics 33.6 (2012): 1599.

[16] Winnie W.Y. Lau, Grace Ngai, Stephen C.F. Chan, and Joey C.Y. Cheung. 2009. Learning programming through fashion and design: a pilot summer course in wearable computing for middle school students. SIGCSE Bull. 41, 1 (March 2009), 504-508.

[17] Ristov, Sashko, Milosh Stolikj, and Nevena Ackovska. "Awakening curiosityHardware education for computer science students." MIPRO, 2011 Proceedings of the 34th International Convention. IEEE, 2011.