T. Hirashima et al. (Eds.) (2011). Proceedings of the 19th International Conference on Computers in Education. Chiang Mai, Thailand: Asia-Pacific Society for Computers in Education

Blogics! A Learning Tool for Enabling Wearable Computing Modules for Beginners

Eduardo VELLOSO[a*], Denise FILIPPO[b] & Hugo FUKS[c]
[a]*School of Computing and Communications, Lancaster University, UK*
[b]*Superior School of Industrial Design, State University of Rio de Janeiro, Brazil*
[c]*Department of Informatics, Pontifical Catholic University of Rio de Janeiro, Brazil*
*e.velloso@lancaster.ac.uk

Abstract: This paper describes the implementation and evaluation of Blogics!, a software system for the learning of logic diagrams aimed at beginners and non-specialists. The system was developed in order to make it viable a Physical Computing module for first year Engineering students with no prior knowledge of Electronics, as part of an effort to diminish student attrition rates. The system was evaluated with beginners using a desktop PC and a touch sensitive whiteboard.

Keywords: Logic circuits simulator, attrition, Boolean logic, digital circuit, freshmen

1. Computer Engineering Education

Many students with an apparent aptitude for Computer Science avoid choosing it as a major [1]. One of the reasons for that may be that students have an incorrect or no perception at all of what it is like to be a computer scientist: most imagine computer scientists sitting in front of computers and programming all day. Therefore, a major challenge in Computer Science and Engineering education and in STEM fields in general is how to motivate these students before and after they enrol in a Computing related major.

Furthermore, many of the ones that do enter Computer Science don't finish the course. Numbers for Computer Science attrition rates in the U.S.A in the first two years of the course vary from 20% to 60% [1]. Women are especially affected by this. Not only men far outnumber women in Computer Science by almost a 4 to 1 ratio [2], the retention rates for women are also smaller than for men [3,4].

Some initiatives to tackle this problem include, among others, the Storytelling Alice Programming Environment [5], the LEGO Mindstorms robotics kit [6] and the Scratch visual programming language [7]. Research shows that Wearable Computing and e-textiles workshops are effective in introducing programming and electronics skills to beginners [8, 9]. These are fundamental skills for a Computer Engineer that need to be learned and improved since the very beginning of the course. Moreover, Lau et al. [9] suggests that wearable computing is exciting and inspiring to students, it can motivate both boys and girls and it can let students explore their creativity while they learn about electronics and programming.

Mahmoud [10] suggests some measures to improve Computer Science education that a Wearable Computing course could implement: it offers a cross-disciplinary content, because the student must consider Electronics, Programming and Fashion Design to build a prototype; it can help fix Computer Science's image by inserting it in the fashion world; it can help increase women's enrolment in Computer Science and, if well designed and taught, a module on Wearable Computing can be fun. That is why we decided to take this approach and create a Wearable Computing course for first year Computer Engineering students at the Pontifical Catholic University of Rio de Janeiro.

Unfortunately, first year students get to University with little or no prior knowledge of logic gates and digital circuits. Moreover, traditional simulation tools, such as Proteus [11] and LabVIEW [12], are great for simulating a wide variety of components and integrated circuits, but they can be rather cumbersome for the beginner. That is why physical computing and embedded systems modules are usually offered after they take other modules on Basic Electronics and Logic [13, 14].

Therefore, although there are teaching approaches of minimizing student attrition, such as collaborative hands-on projects, there is also the need to develop tools to support these activities. That is why, in order to make viable this Wearable Computing module for first year students in our University, we developed Blogics!, a digital circuit simulator especially aimed at beginners. It emphasizes real world elements – sensors, actuators and even the different variables of the environment, such as luminosity and humidity – in the construction of digital circuits for enhancing beginners' understanding of the connections between real world and the circuits they will work on.

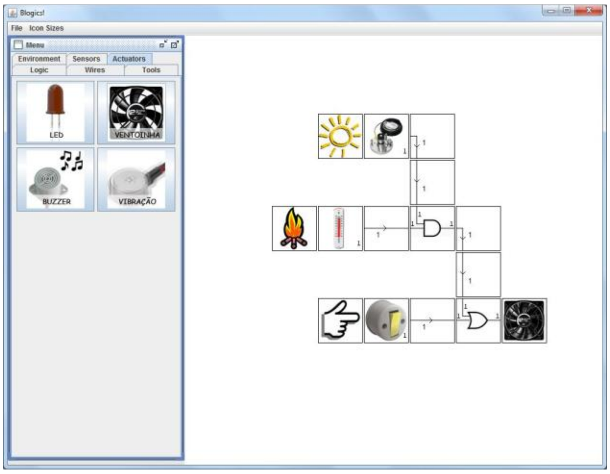

Fig. 1. A circuit built with Blogics!.

2. **Blogics!**

Blogics! is a logic circuit simulation software in which logic gates, wires, sensors and actuators are created and manipulated as moving blocks, as to make the interaction easier for users from different areas other than STEM, such as designers and artists as well as beginners in Electronics, such as first year Computer Engineering students.

The goal of the system is to offer the basic resources for a didactic logic circuit design experience and not a sophisticated simulation of electronic circuits and systems. In its current version it simulates logic gates (AND, OR, NOT, NAND, NOR, XOR, XNOR), truth values (1, 0, blank), sensors (button, switch, presence, humidity, luminosity, temperature), environment variables that alter sensors' values (finger, room with and without people, dry and humid weather, day and night, hot and cold), actuators (LED, buzzer, vibration motor, fan) and wires (Figure 1). A digital circuit is built by arranging and connecting blocks over the screen. Sensors and truth values are used as inputs for the circuit and actuators will turn on and off accordingly. When environment variables are connected to sensors, the circuit will immediately present the results. For instance, if a "Hot" block is connected to a "Thermometer" block, the latter's output value will be "1", while if a "Cold" block is connected to it, its output value will be "0". When sensors and actuators have their values changed, their corresponding image is also altered to reflect these changes and the corresponding Boolean value is indicated on the block. For instance, when LED is on the image shows a bright red LED, but when it is off, it shows a dimmed LED. This paradigm was chosen in order to emphasize a systemic approach to the circuit building process: the student must figure out which are the inputs, outputs and how they interact with and within the environment. Simulation occurs in real time. The user interface is designed for the system to be used in a touch sensitive surface. Therefore, the user only needs to click, drag and drop blocks to build a circuit. It doesn't require right clicks, double clicks and text entry since these actions are rather cumbersome in a touch sensitive display. Its design followed touchscreen interface design guidelines proposed by [15].

3. Case Study

Given that an obstacle for the implementation of a first year Wearable Computing module is the gap between the little of Electronics and Logic that students know and the minimum that they need to understand in order to build a circuit, we wanted to check whether Blogics! could help bridge this gap.

In order to evaluate the effectiveness of the software we conducted a case study with students of a graduate course at the Superior School of Industrial Design of the Rio de Janeiro State University (ESDI/UERJ) in the beginning of a Physical Computing module. At the start of the case study, the students had a single lecture on logic gates, in which they learned how to translate natural language sentences to formal logic statements and to design the corresponding circuits using the AND, OR and NOT gates.

One week later, the case study proceeded with a revision of the previous lecture, a presentation of the software, a pre-activity questionnaire, a pre-test, a learning activity using Blogics!, a post-test and, finally, a post-activity questionnaire. The learning activity was elaborated to teach them the XOR logic gate, which was unknown to all of the students and wasn’t presented in the previous lecture.

It started with a 15 minute revision of the previous lecture on logic gates, followed by a 10 minute introduction to the tool. Afterwards, the students answered a questionnaire so we could figure out their profiles for group formation purposes. At this point, they took a 10 minute pre-test to assess their current knowledge on the subject. It included an exercise for which the solution could easily be found if they had known the XOR truth table.

The class of 8 students was divided into 4 pairs, based on their pre-test score (good scores paired with bad scores) and their answers on the questionnaire. The members of 3 of the 4 pairs didn´t know each other before the activity. Two of these pairs were moved to a separate room equipped with a desktop PC computer and the other 2 pairs to one with a SMART Board 680 Interactive Whiteboard. This is a 77” single touch sensitive vertical surface widely used in interactive classrooms [16].

Each pair of students received 2 exercises to be solved in 20 minutes. In the first one, they were to use Blogics! in order to learn the truth table of the XOR gate. In the second one, a project specification was presented and they had to build the corresponding circuit. This problem could be solved using a wide variety of gates, but if the students used the gate they had just simulated, they could find a very short solution to the problem.

After the exercise, students were asked to take a post-test, with the same level of difficulty of the pre-test, to check whether their performance would improve and whether they would use the XOR gate in the post-test. Finally, the students received by email a link to a questionnaire asking about their experience with the activity to be returned within a week.

4. Results

Seven out of the eight students answered the questionnaire and gave feedback about the software including: “The tool is very simple and objective.”, “The tool is very effective”, “It was fun using Blogics!”, “Working with graphic elements helped a lot to understand the exercise”, “What I liked: Visualization of how programming can be applied in real life.” Mixed responses included “Blogics! enables us to work on a trial and error basis, so we could get the task right, but we not necessarily understood what we were doing”.

Students commented on the software featuring sensors and actuators: “The idea to separate sensors, actuators and logic concepts will help a lot when the time comes to translate it to code. This way, we could work separately the inputs, the outputs and the internal logic”. Several responses also commented on the possibility of seeing the results of the system in real time, during the construction process: “It made the resolution of the exercise easier because we could immediately see the implication of using one logic gate or other. The result of each action was immediate”.

The exercise with Blogics! made it possible for them to contextualize what they were studying. One student observed that the exercise clarified the students’ idea about what they were studying: “Before [the exercise], we were intrigued about the Arduino [microcontroller prototyping board] and its possibilities (…). Now we know that the knowledge about digital logic, Boolean operations and logic circuit construction is much more important than the potentialities of a small equipment that does nothing more than process inputs and outputs”. Another student stated that he gained “a better understating of the new technology that we are studying in the Physical Design module.”

The usage of a graphic representation of electronic elements was also commented: “Working with graphic elements and pictures help a lot to understand the exercise”.

The negative points they commented on the software were related to the interface: “the need for a hand block to activate a switch”, “the bin [block]”, “we could have the option to select several blocks and move them at once”.

When asked about what they learned students showed an increase in their knowledge on the subject: “I learned how the XOR logic gate works [...]. Besides, I could visualize how Boolean logic works inside a computer, especially when I visualized the result of each action I made, I had an immediate feedback of what I was doing”.

The most promising results appeared when we compared the students’ performance in the pre-test against their performance in the post-test. Most students improved their performance, while two kept their scores and one actually got a worse result. The pair that kept their scores spent most of the time (10:35’’ of the 20:00’’ they had) trying to solve the exercises on paper, instead of using the tool to figure out the XOR truth table. Without the software, they were not able to build the XOR truth table and skipped directly to the second exercise. Even when they started to use the interactive board, they tried to solve the second exercise first by hand. Hence, they were the only group that didn`t use the XOR gate to solve the exercise. The others started building the circuits with Blogics! straight away. The student that got a worse result used a desktop PC, but he didn’t hold the mouse during the activity. We posit that the actual interaction with the software is an important factor in the student’s learning process. Remarkable results that indicate the success of the activity were the students who improved their scores significantly from a 5 to a 10 and from a 0 to an 8, respectively. Unfortunately, only 1 of the 6 students that used the XOR gate in the exercise did use the XOR gate in the post-test. The bright side is that they figured out a way of solving it using AND, OR and NOT gates, differently from the pre-test.

These results confirm our hypothesis that a software like Blogics! can support beginners in grasping Digital Circuits concepts. Most of the students that actually used the tool to build the circuits figured out the XOR truth table and improved their scores.

5. Conclusion

Student attrition rates are a huge problem in Computer Science education around the world. In order to try to minimize this problem in our University, we decided to offer a module on Computing for first year students. In order to make this course viable, we developed Blogics!, a software tool designed to mitigate the difficulties that beginners and non-specialists have in their first contact with digital circuits.

In order to achieve this, the software was built to support them with a concise tool that brings elements of the real world into the circuit design process: not only by using sensors and actuators, but also bringing the environment itself in the equation as variables such as hot/cold, high/low humidity and presence/absence of a person in a room.

The goal of this work was to investigate this tool and its educational potential. Blogics! was evaluated with beginners in the subject of Digital Circuits and we found promising results: students seemed to bridge the gap between the real world and the logic circuit's abstraction and they were able to freely explore the logic in a trial and error way with immediate feedback. Hence, this study showed that a software with Blogics!' features has can be used as a didactic tool for introductory Electronics courses. This is a small but important part in the effort to empower beginners and diminish attrition rates for both genders in Computer Science and Engineering. Further insights on our approach to minimize attrition rates will be gained in the long term, once the Wearable Computing module is implemented.

References

[1] R. H. Sloan and P. Troy, CS 0.5: a better approach to introductory computer science for majors. In Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education (Portland, OR, USA, March 12 - 15, 2008). SIGCSE '08. ACM, New York, NY, 271-275.

[2] C. Hill, C. Corbett, and A. St. Rose. Why so few? Women in Science, Technology, Engineering, and Mathematics. Washington, DC: AAUW Research Report, 2010.

[3] J. Margolis and A. Fisher. Unlocking the Clubhouse: Women in Computing. The MIT Press, Cambridge, MA, 2002.

[4] R. Powell, Improving the persistence of first-year undergraduate women in Computer Science. Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education. Portland, OR, 2008.

[5] C. Kelleher and R. Pausch, Using Storytelling to Motivate Programming. Communications of the ACM, 50, 7 (July 2007), 59-64.

[6] F. Klassner, "A case study of LEGO MindStorms' suitability for artificial intelligence and robotics courses at the college level," in Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education (2002).

[7] M. Resnick, J. Maloney, A. Monroy-Hernandez, N. Rusk, E. Eastmond, K. Brennan, et al., Scratch: Programming for Everyone. Communications of the ACM.

[8] 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 (CHI '08), Florence, Italy, 2008, pp. 423-432.

[9] W. W. Y. Lau, G. Ngai, S. C. F. Chan, and J. C. Y. Cheung, Learning programming through fashion and design: A pilot summer course in wearable computing for middle school students. In Proceedings of SIGCSE '09, Chattanooga, Tennessee, March 2009.

[10] Q.H. Mahmoud, "Revitalizing Computer Science Education", Computer Magazine, May 2005.

[11] Proteus VSM Simulator. http://www.labcenter.co.uk , last accessed 28 September, 2010.

[12] NI LabVIEW. http://www.ni.com/labview/ last accessed 28 September, 2010.

[13] Embedded System Design Course Description, University of Colorado, http://ecee.colorado.edu/~mcclurel/syllabus.html , last accessed 28 September, 2010.

[14] Syllabus for EE 459Lx Embedded Systems Design Laboratory, University of Southern California, http://ee.usc.edu/academics/class_webpages/ee459/documents/EE459_Syllabus.pdf, last accessed 28 September, 2010.

[15] C. M. Brown, Human-Computer Design Guidelines, United Kingdom: Intellect, 1998, pp. 135-137.

[16] SMART Technologies, http://smarttech.com/ , last accessed 28 September, 2010.