Int. J. Engng Ed. Vol. 23, No. 5, pp. 868–873, 2007
Printed in Great Britain.

0949-149X/91 $3.00+0.00

Teaching Software Engineering Principles Using Robolab and Lego Mindstorms*

RENA HIXON
Wichita Homeschool Warriors Lego Robotics Club, USA. E-mail: erhixon@swbell.net

Engineering can be made real and enjoyable to young children while providing opportunities to teach engineering applications. A robotics camp was developed to teach general programming concepts to elementary and middle school children using Lego Mindstorms kits. The camp could easily be expanded to teach students in high school. In recent years, engineering has been taught in very few K–12 schools causing a problem in recruiting students for engineering at the college level. Ideas for promoting engineering include training teachers at the K–12 grades to teach engineering and involving parents at this age level to follow through to graduation with training [1]. The curriculum developed for the weeklong camp could be used by fourth through eighth grade teachers to acquaint students with programming skills needed for computer engineering in high school and college. Since children of all ages enjoy playing with robots, the Team Challenge (Lego Mindstorms) kit developed by Lego Educational Division is an excellent teaching tool. Robolab, developed by Tufts University, can be used to teach computer engineering principles to fairly young students. Students starting at age ten, eleven or beyond can start to understand and enjoy software development using these kits. A weeklong camp using Team Challenge kits with Robolab was developed to teach children of these ages programming concepts that could be used in all programme design. An advantage of Robolab for this age group is the icon-based language, making it easier to understand than a written programming language.

Keywords: middle School education; Robolab; Lego Mindstorms

INTRODUCTION

EDUCATING AND GENERATING ENJOYMENT in young children for engineering, science and mathematics can promote career choices in these fields. Too often, children lose interest in these subjects during younger years, preventing them from pursuing these topics in college and beyond. More students need to be interested in engineering and science to enhance the United States economy in the future. To interest students in engineering in college, there is the need for more educational opportunities at the precollege level to prepare students to enter these fields at the college level [2]. To increase the interest among United States students, parents and teachers must be trained to teach engineering and science.

Companies, such as Lego Education, are developing products that encourage children in younger grades to have an interest in science and engineering. Even though the products are available, many of the teachers are not trained to teach science and engineering topics [3]. Tools and curriculum need to be developed for use by K–12 teachers to interest young children in engineering principles. Robots can make learning enjoyable and encourage children to continue learning.

As the curriculum standards for Kansas third and fourth grade science show, teachers are encouraged to teach science. They are expected to provide students with 'a variety of educational experiences which involve science and technology' [4]. They want the student to 'begin to understand the design process, which includes this general sequence: state the problem, the design, and the solution' [4]. There are two benchmarks that are associated with this standard. The first is that a student works with a design problem and the second is that a 'student will apply their understanding about science and technology'. [4] For fifth through seventh graders, an additional benchmark is that the student 'will develop understandings of the similarities, differences, and relationships in science and technology'. [4] Teaching programming in fourth grade and beyond can provide these opportunities.

Born out of the desire to teach that engineering is a process and not guess work, a homeschool robotics club formed in the autumn of 2004. The objective of the club was to promote learning and interest in engineering using the Team Challenge kits with Robolab software. Reference 10 gives an idea of the type of activities conducted by the club. Due to lack of computers and time, in-depth programming concepts were unable to be taught. From this came the interest in teaching an advanced programming camp to expose these children that were already being taught engineering principles to programming concepts.

The concepts outlined in this paper are targeted towards fourth through eighth grade students and were developed by a software design engineer. The

* Accepted 10 August 2007.

goal of the camp is to use Robolab to teach programming principles that are not generally taught at this level. In order to develop working programs, a student is required to understand how the robot operates and to state the problem before designing the program. Not only are the principles of computer engineering used for program development, but the basic engineering process is necessary to create a working program.

Objectives

The overall objective of the camp was to promote interest and teach programming features and advanced concepts used in all programming languages. The camp indicated that students as young as ten could understand the concepts, use them and continue to build robots and program.

All programming languages require a knowledge of loops, statements, subroutines, variables and interrupts. More advanced techniques used in programming languages are suspending a program until another event occurs and task splits. In Table 1, programming terms used in computer science are given and the equivalent terms in Robolab are shown.

At most universities, an introductory computer science class will teach an introduction to high-level programming. This assumes that most high school students have not had this training. In the standards developed by IEEE Computer Society for teaching computer engineering [3], an entire section is devoted to programming fundamentals and the courses that should be offered in this area at the college level. One course is a history course teaching reasons for programming. Students using Robolab with Lego Mindstorm kits have experience using programs to control robots. Another college course is designed to teach high-level programming emphasizing many of the concepts taught to the children in this summer course.

The Lego Company's Educational Division [5] sells a Team Challenge kit with Robolab [6] software, written by Tufts University. Two groups of individuals use these kits; precollege science teachers and professors in engineering. The precollege science teachers have a background in math and science, but have not had practical experience. The professors in engineering use these same kits to teach college students [7] [8]. Some of the same people developing college curricula are providing similar experiences for K–12 education [9], but the resources available for educators in the classroom at this level are minimal. Therefore, a curriculum for elementary educators needs to be developed so they can teach concepts that they themselves may not have learned. This material needs to be interesting and fun, but still informative. Additionally, students need to know that engineering is more than just guessing.

Process

Robolab is an icon-based programming language developed by Tufts University on top of Labview, which is used in industry for test equipment. Robolab was developed to control a Lego brick called the RCX that has three outputs that can control devices such as motors and lights and three inputs that can have various sensors attached. Some of the Labview commands are accessible in Robolab. Those commands could be used for teaching more in- depth programming for high school students.

To teach programming, a simple robot needs to be built that can be used for all of the programming concepts that are to be taught. A simple car with motors and a short wheel base is best.

Flowcharts are good tools to help facilitate younger students to learn programming. Simple blocks describe the actions that are to take place and the lines and arrows demonstrate well where controls in the program should be placed. Using the flowcharts with the icon-based language make the programs easy to write.

The progression of teaching concepts is important for helping students to understand programming. The first thing to teach the students is that programs only run once. This can be done by using Robolab commands that turn on motors and turn them off again. Students can observe by running the program on the robot that the program only executes once.

Once the students understand that programs only run through once, then loops should be introduced—conditional and unconditional loops. These are necessary for more complicated programs to be written. Unconditional loops produce programs that run forever until the power is shut off to the robot. Conditional loops check for some action, such as a touch sensor to be pressed or a light value to change.

Program suspension is a more complicated concept to teach in higher level languages, but is

Table 1. Computer engineering terms and related Robolab terms

Programming Terms	Robolab Terms
1. Loops, conditional and unconditional	1. Loops
2. If statements	2. Forks
3. Program suspension	3. Wait for
4. Subroutines	4. Subroutines
5. Variables	5. Containers
6. Multi-tasking	6. Task splits
7. Interrupts	7. Event Programming

quite simple to teach with Robolab. When the Robolab 'wait for' commands are used with the robot, it is easy to observe that nothing else is happening until the condition, such as a light sensor value or touch sensor pressed, is met. Observing the behaviour of the robot helps the students to understand the program.

The students must be taught to write subroutines and how to implement them. In Robolab, this is much simpler than in higher level languages. Each subroutine is assigned a number indicating what subroutine it is. This number is used to execute the subroutine.

Using containers in Robolab is a more complex programming concept because containers can be used in all of the previously taught and future concepts. The use of the word containers is helpful because it helps the children understand the idea of holding something. Introducing the term 'variables' can help the students understand the values in the containers can be changed while introducing them to terms used in later programming classes.

Multi-tasking is taught by giving the students ideas that they may already understand on computers, such as using a word processor while searching the Internet. This helps students understand the term and at the same time relates it to computer technology. Examples are given to the students by a robot playing a tune at the same time as it is checking a sensor. Without multi-tasking, the robot would either check the sensor or play a tune.

Interrupts are best taught by giving several programming examples using different sensors and letting the students observe what is happening on the robot. Explaining the functioning of the programs as the robot executes helps the students to grasp these more complex ideas.

The use of a robot to teach the concepts makes learning enjoyable for the students and they receive immediate feedback on whether the robot works or not. When they see something work, many times it gives children ideas of their own to try.

Implementation and observations

In the summer of 2005, a pilot group of nine children, with ages ranging from nine to thirteen, all of them schooled at home, were invited to attend the camp. The children attended for five days, three hours a day. The invited ones already had experience using Robolab and, except for three, had competed in a local robotics competition. The students worked in groups of two, sharing a computer. On the first four days, instruction was given; on the fifth day, the children were given two different challenges to attempt.

From previously working with these children, it was apparent that they used a small subset of the features that were available in Robolab and did not totally understand the programs that they had written. The objective was first to make sure they understood how simple programs worked and then teach advanced programming techniques using hands-on examples. One of the robots designed for the competition could not be turned off with the on/off button. Their solution was to turn off the power because they did not know why the on/off button did not work. After the students ran the first program given in the camp, they commented that they now understood why the on/off switch would not stop the program.

Before the camp, several of the students had already been using Robolab. Few, if any, of the students before the camp understood loops which are a basic programming concept in all languages. After introducing loops on the first day, most of the students grasped the importance of loops and how they worked.

In order to teach the material, the specific concept was taught first. It was demonstrated with an example on the robot. Afterwards, the students were given a program to write themselves, test it and get it working. When necessary, material was reviewed. If the new concept could be combined with what was already taught, more examples and programs were given to include both concepts.

One of the exercises given to the students was to read two different light values and calculate a light value with which to follow a line. The students were instructed to read a light value, sound a tone and physically move the robot to read the second value. One of the students put the robot down and left it instead of physically moving it to read the second value. The robot had been programmed to turn to read the second value, instead of the student moving the robot!

On the last day, the students were given two challenges. One of the challenges was a duplicate from a local robotics competition which was partly composed of line following at the beginning, detecting a touch at the end of the line and turning around and following the line in a different direction. The second challenge required using a light sensor to determine different values of dots on paper and to count the number of dots lined up on the page. They were allowed to choose which challenge they wanted to do. The children had to use different concepts presented during the week.

Most of the children chose to do the duplicated challenge. It was quite surprising when a ten-year-old chose to use an interrupt to trigger a touch condition! Many college students find this concept challenging. To the student, he was simply using event programming—a technique he had been taught earlier in the week.

For competition, the challenges are released six weeks before the event. The challenge used for this camp had taken the children all six weeks to complete. Two of the groups completed the first third of the challenge on the last day of camp. One of the groups chose to experiment on the first third of the challenge using two different programming approaches. They used the normal 'wait for' and then event triggering for the touch sensor.

The dot counting challenge needed to be better designed and the children had problems with it. They discovered that the robot had to go fairly slow and straight to detect and count the dots. The students had the program working when the robot would go straight. Using different coloured lines would be a more effective challenge. This is a very good course for the students to learn about containers because they are necessary to accomplish the task.

Assessment

The original camp was not designed to assess what the students learned. The idea was to teach fourth to eighth graders skills to help them make more powerful robots. An assessment was given to the parents to help determine the immediate effectiveness of the camp, but the real assessment would be obtained by observing these kids work with their robots in the future. Since one of the goals is to teach kids that programming can be fun so they will want to pursue it in the future, the courses these children choose in high school and later on in college would help determine the overall effectiveness of the camp.

The survey was given to the parents about a month after the camp. They were asked three questions:

1) Was the camp a positive educational experience for your child?
2) Has the child been using the skills he developed since camp?
3) Did this experience give them a more positive view of engineering?

They were asked to rate each question on a scale from one to five with five being positive and one being negative. The average results are shown in Table 2.

During discussion with the parents, most of them commented that the reason that their child had not been programming was due to a lack of time during the summer. One parent said their child had not done any programming because one of his major Lego components for the robot had been broken since camp. The other parent who answered the question with a one said they had been out of town or involved in other activities since the camp. All of the parents expressed complete satisfaction with the camp and were excited to have skills taught to their students that they did not feel qualified to teach.

At one point during the camp, the students were asked if they felt they had learned much during the week. All of the students responded positively. One student commented that now that he had been exposed to it, he felt he would more readily use the concepts and understand better how to do so. Figure 1 shows the students at the camp.

Table 2. Results from parent survey

Question #1	Question #2	Question #3
5	2.6	4.8

After the camp, a few of the students participated in a Lego robotics club. They were building new robots and using the programming concepts used in the camp. They helped to teach the programming concepts to new students who did not have the experience of the programming camp. The robots provide an incentive for the children to continue programming and provide them with a good basis for studying software engineering in later years.

Additional questions that could be asked to determine the effectiveness of the camp would be:

1) Did the students find the camp enjoyable and want to use the new skills learned?
2) Do the children continue building robots and using the programming language?
3) Are they developing ideas of their own and using the programming concepts taught?
4) Are they able to teach others what they learned?

Summary

Even if the students did not fully understand all of the ideas, they were exposed to many programming language concepts. The kids enjoyed being able to make the robot respond and knew it was accomplished by programming. When encountering these same concepts later, the students will already have some understanding of them.

Engineering can be made real and enjoyable to young children while preparing them for programming in high school and college. These children can be shown that engineering is a process and guessing is not the best way to approach a design problem. The children used the key programming concepts of loops, if statements, variables, interrupts, subroutines, program suspension and multitasking to accomplish various tasks for the robots. The children learned some complex programming concepts that even college students have problems grasping. By requiring the students to state the problem before designing a program, they grasp the basic engineering process.

Fig. 1. Students participating in robotics camp.

By the end of the week, the majority of the students were able to use the programming concepts that had been presented. The youngest two students (one nine and one ten-years-old) had learned more about programming concepts, but did not grasp all of the programming details. Both of these two younger students are much more interested in building than programming. The rest of the students appeared to understand and be able to use the majority of the programming concepts. All of the children expressed enjoyment in the camp, proving that students could find enjoyment in this type of programme.

By giving the students flowcharts with each program and encouraging them to draw their own, the children were shown that there is a process to be followed, rather than just guessing about how to write programs. The students (or even the precollege teachers) may not know the computer science terms for their programming concepts, but they understand how to methodically approach a design problem.

An advantage of teaching Robolab with Team Challenge kits is the ability for the students to immediately see the results of their programs. Embedded system programs control hardware and the students using Lego Mindstorms are getting to control motors and lights and read in values similar to an embedded system. It is also a good tool for teaching younger students programming because the icons make it easier for younger students to understand. The concepts are no less complex because they are icon-based, but easier for younger students to visualize. The students do not have to be concerned with the syntax of a written language, but can learn advanced programming ideas. The students attending this camp are given skills that can be used for future work in a programming field. They have the opportunity to use Robolab in robotics competitions for fourth through eight graders sponsored by First Lego League [12] across the United States and in state competitions such as those held annually at Wichita State University [13].

The students in this camp were taught all of the programming concepts presented in Table 1 which are not commonly taught to elementary grade students. These cover basic and some advanced programming principles that are in all programming languages. Most K–12 teachers do not have programming experience that prepares them to teach children at this level. This type of curriculum is necessary to prepare the elementary and middle school children for classes in high school and to instill an interest in computer engineering. The children exposed to this teaching in younger years are more likely to become the engineers to enhance the economy of the United States in the future.

Based on presentations at a Robolab Conference in August, 2005 [14], the lack of material available and conversations with Lego Education, the material presented is unique to this age group at this teaching level. Eric Wang [15] produced a textbook for Robolab targeted for college. Since there are no other books available for younger students, his book has been used by K–12. The curriculum used for the camp was specifically designed for fourth to eighth grade students.

REFERENCES

1. Kansas Department of Education, March 2006 http://www.ksde.org/outcomes/sciencestd3t4.doc
2. Agents of Change: Achieving Diversity in Electrical and Computer Engineering Research and Education, http://www.ecedha.org/reports/AoC_Report.htm#_Toc55038925 National Science Foundation, April 2006.
3. Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering http://www.eng.auburn.edu/ece/CCCE/CCCE FinalReport-2004Dec12.pdf, Joint Task Force on Computer Engineering Curricula, April 2006.
4. America's Pressing Challenge—Building a Stronger Foundation, http://www.nsf.gov/statistics/nsb0602/ National Science Foundation, April 2006.
5. http://www.legoeducation.com Lego Education, August 2005.
6. http://www.ceeo.tufts.edu/ robolabatceeo Robolab@CEEO, August 2005.
7. Chris Rogers and Merredith Portsmore, Data acquisition in the dorm room: Teaching experimentation techniques using LEGO Materials, *Proceedings of the 2001 ASEE Annual Conference*, Albuquerque, New Mexico, 2001.
8. Philip Lau, et al. LEGO Robotics in Engineering, *Proceedings of the 2001 ASEE Annual Conference*, Albuquerque, New Mexico, 2001.
9. Chris Rogers and Merredith Portsmore, Bringing Engineering to Elementary School, *J. STEM Educ.*. **4**, 2004.
10. Rena Hixon, Engineering Education for Young Students Using Lego Mindstorms, Robolab Conference, Austin, TX, August 2005.
11. www.ceeo.tufts.edu/robolabatceeo/Resources/documentation/ROBOLAB_Reference_Guide.pdf, Robolab@CEEO, August 2005.
12. http://www.firstlegoleague.org/default.aspx?pid=70 First Lego League, May 2006.
13. Robotics in the Classroom, http://education.wichita.edu/mindstorms/challenge.html Wichita State University, May 2006.
14. http://130.64.87.22/services/conferences/robolabconference/ International Robolab Conference, May 2006.
15. Eric Wang, Engineering with Lego Bricks and Robolab, College House Enterprises, LLC, 2004.

Rena Hixon graduated from Wichita State University with a Ph.D. and M.S. in Electrical Engineering. She received her undergraduate degree in Computer Science from the University of Missouri-Rolla. She was a software design engineer for nine years and worked as a MVS systems programmer for two years. She taught C, C++, and assembly programming for eight years. Currently, she is schooling her three children at home. She started a homeschool robotics club in 2004 for fourth to eighth grade students and co-teaches the group with her husband. She does some independent program consulting on the side.