World Transactions on Engineering and Technology Education
Vol.10, No.3, 2012

Use of a robot simulator to enhance robotic peer mentoring programme in pre-tertiary schools

Hugh Considine, Ivan Lee, Andrew Nafalski & Zorica Nedic

University of South Australia
Adelaide, Australia

ABSTRACT: The project described in this article was an enhancement of the Robotic Peer Mentoring programme (RPM) successfully implemented by the University of South Australia (UniSA) for the past decade that has exposed school students to electronic engineering, computer science and robotics. Primary and secondary school students were asked to complete some simple programming exercises using a developed robot simulator and were then surveyed to find out if they had enjoyed programming and using the simulator. Overall, the results indicated that students had found the simulator stimulating, and that most students who had used the simulator were willing to consider a career in computer science or in other robotics related fields.

INTRODUCTION

Encouraging school students to study information technology (IT) and engineering related programmes at university level is an ongoing and difficult challenge. One approach is to offer school students a chance to learn some basic technical skills, and then to give them the chance to solve simple challenges using these skills. This builds the students' confidence in their own abilities, and shows them that they can take charge of modern technology rather than being passive users.

Previous studies have found the use of robots to be beneficial. In the paper by Burhans, D. et al, the authors used Lego Mindstorms robots in introductory computer science classes [1]. When students were surveyed, they had enjoyed using the robots and found that they understood theoretical concepts better.

Prior work at the University of South Australia, described later in this article, has also had positive results. Secondary school students were given robotic kits to assemble and program. Courseware and kits were developed to teach students basic electronics and programming. In this study, a robot simulator was used to give students an exposure to robotics in a programming/simulation environment.

THE BACKGROUND

Robotics Peer Mentoring

To attract school students to study engineering and IT related subjects, the University of South Australia conducts a robotic peer mentoring programme (RPM) in which current university students of UniSA, the University of Adelaide and the Tertiary and Furthering Education (TAFE) work with school teachers to teach a basic robotics course [2][3]. The RPM project was unique in Australia and has attracted the Australian Engineering Excellence Award in 2003 [4]. In the course, school students spend around eight weeks building the Envirocar robot, which they then program using the Great Cow Graphical BASIC programming environment [5]. This gives the school students a sample of electronic and software engineering, which they may then choose to study further at a tertiary level. In the past, the RPM programme has only involved school students in years 9, 10 and 11. However, the simulator was also used with students in year 7 at a primary school.

Envirocar Robot

UniSA developed the robot kit used for the mentoring programme, known as the Envirocar, shown in Figure 1. It is a microcontroller based racing car kit, intended to provide a simple introduction to electronics. Students receive a kit of parts, and are required to use a soldering iron and other tools to create the finished car. The Envirocar is based on the Microchip PIC 12F675 microcontroller. In addition to the microcontroller, the car has:

- Two motors, which allow it to drive forward, left and right;
- A light sensor, which allows it to detect lines marked on the ground;
- Headlights, which aid the function of the light sensor and which can be used to give some visual output;
- A user programmable button that is normally used to start and stop programmes.

Students can program the Envirocar to perform a predefined sequence of movements or to react to input from the light sensor and a button.

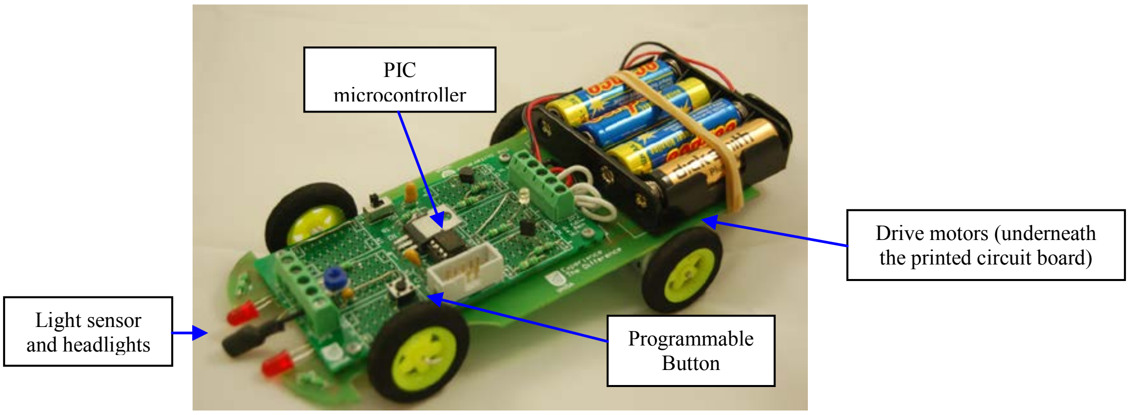
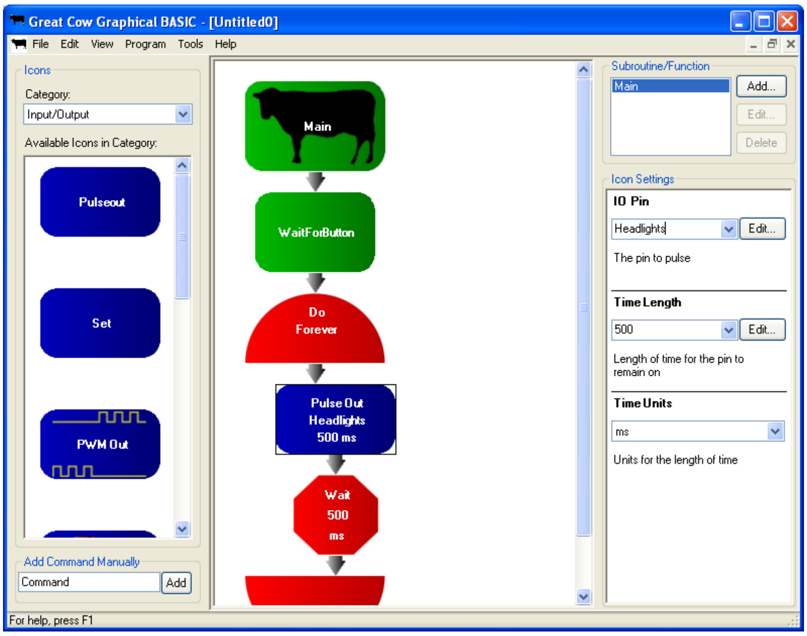

PIC microcontroller
Drive motors (underneath the printed circuit board)
Light sensor and headlights
Programmable Button

Figure 1: Envirocar.

Graphical Programming: Great Cow Graphical BASIC

Great Cow Graphical BASIC is an open source in-house developed software package that allows robots to be programmed using icons. A sample of a program is shown in Figure 2. Each icon represents a single command in the underlying Great Cow BASIC language. Students can drag and drop icons into a sequence, and then alter the settings of the icons to produce a program. When the students click on the Compile and Download menu, the program is converted into the BASIC code. This is then compiled to produce a machine code and sent it to the microcontroller.



IO Pin
Headlights
The pin to pulse

Time Length
500
Length of time for the pin to remain on

Time Units
ms
Units for the length of time

Figure 2: Great Cow Graphical BASIC.

Motivation for a Robot Simulation

Some secondary school students are more interested in programming than in electronics. However, to complete any programming, they must first build a robot. Some students do not want to build a robot, or are unable to do so in the time allocated to the robotics class. Another issue encountered is the limited budget of some schools to provide a robot hardware development kit to each participating student.

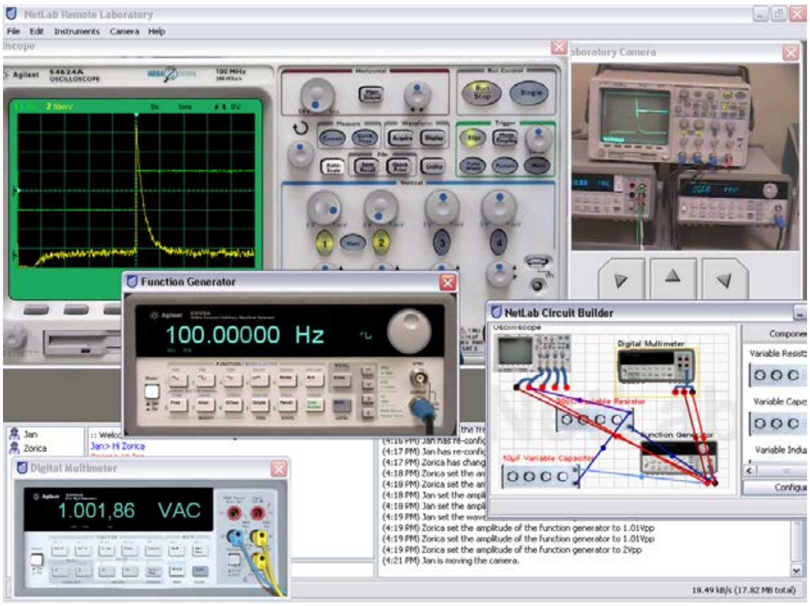
One possible way around these problems is to use a robot simulator. This allows programming to be completed without a physical robot - so students can focus more on software if their interests lie there rather than in electronic construction. The use of a simulator could reduce the amount of expensive hardware required - rather than buying a class set of robots, the school could buy a smaller number of physical robots, and use a simulated version of the robot for most of the training exercises.

The simulator can also help to demonstrate how two otherwise identical robots can behave in a slightly different way. For example, some motors will have slightly more power than others, or some light sensors will give slightly different readings when presented with the same light level. If a student only uses one robot, he or she may not realise this. The simulator presents a second robot, and this can demonstrate to students the importance of not assuming that the same program will always behave identically.

REMOTE LABORATORIES

Remote laboratories combine simulation (the prominent part of is the graphical user interface GUI) with remotely operated hardware, thus resembling the concept used in the RPM where students use both real hardware and simulation. The major difference with remote laboratories is that the hardware and the students are in different physical locations. The students can control the hardware/software combo from any place in the world with Internet access. Remote laboratories gained popularity with the rapid uptake of the Internet in the mid to late 1990s. These laboratories enhance the students' learning experience and flexibility of studying, allowing 24/7 access. The students can have access to unique and expensive equipment. The cost of equipment (usually only one set of it is required) and supervisory staff is greatly reduced, as the students work with minimal supervision.

One of the more intuitive and flexible in use is the remote laboratory NetLab developed at the University of South Australia about a decade ago [6][7]. It is an open access laboratory (http://netlab.unisa.edu.au) with a remotely configurable hardware setup, whereby different instruments and variable components can be physically connected together with a click of a mouse. As shown in Figure 3, the NetLab GUI has a number of animated front panels for control of real instruments, a fully controllable camera image and the Circuit Builder for configuring basic electrical circuits. Measurement results can be remotely downloaded and further analysed for a laboratory report.



:: Welco
Jan> Hi Zorica

(4:16 PM) Jan has re-config
(4:17 PM) Jan has re-config
(4:17 PM) Zorica has chang
(4:18 PM) Zorica set the am
(4:18 PM) Zorica set the am
(4:18 PM) Jan set the ampli
(4:18 PM) Jan set the ampli
(4:19 PM) Jan set the wave
(4:19 PM) Zorica set the amplitude of the function generator to 1.01Vpp
(4:19 PM) Zorica set the amplitude of the function generator to 1.01Vpp
(4:19 PM) Zorica set the amplitude of the function generator to 2Vpp
(4:21 PM) Jan is moving the camera.

Figure 3: NetLab GUI.

The remote laboratory was evaluated in some earlier works [8][9]. Most of the students surveyed found the ability to conduct experiments outside of class times helpful, and also enjoyed the reliability of the system. Overall, 90% of the students surveyed answered that the remote laboratory had been at least as effective as the conventional laboratory. This demonstrates that students are happy to work with a virtual interface, so a simulator may be equally effective.

GREAT COW ROBOT SIMULATOR

The Great Cow Robot Simulator was developed for the Robotics Peer Mentoring programme. It is integrated with the Great Cow Graphical BASIC programming system, and simulates the Envirocar robot.

Since it is intended for use by secondary school students with little or no programming experience, the simulator was deliberately kept simple. Students can interact with the simulated robot in the same way as they would with a physical robot. Pressing a button on the image of the simulated robot sends a signal that the button has been pressed to the simulated microcontroller, in exactly the same way as it would on the real robot.

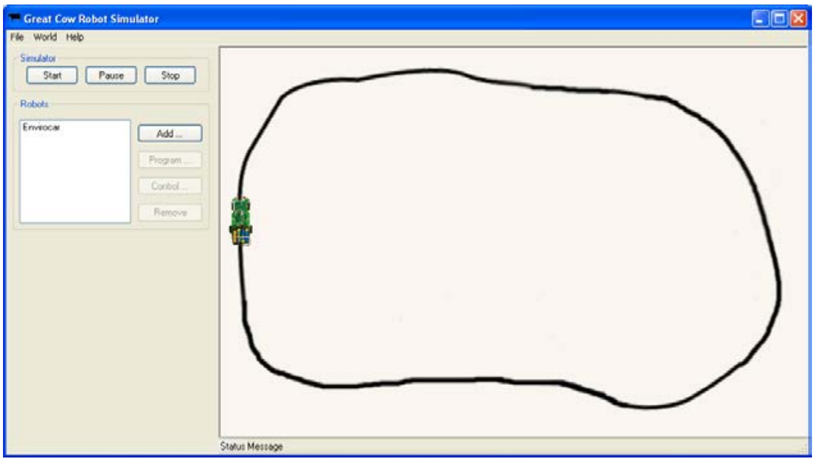


Figure 4: Great Cow Simulator, with the Envirocar robot following a line.

The main intention of the simulator is to show how a program would function on the real Envirocar, so many common debugging features like breakpoints and the ability to modify the program counter have been left out. The simulator shows an entire robot, but it simulates the individual components that make up the robot. This allows it to provide an accurate simulation, and makes it simpler to add more robots to the simulator in the future. Each robot can have one or more microcontroller chips, plus many input and output devices.

Currently, the simulator supports 8 bit PIC microcontrollers, including the 12F675 chip used on the Envirocar. It simulates the data and program memory of the chip, as well as the input and output pins. Machine code programs from Great Cow Graphical BASIC (or another compiler) are loaded into the program memory and then run. The programs then function exactly as they would on a real robot. In addition, the simulator also simulates the following devices:

- Light Emitting Diodes;
- Motors;
- Liquid Crystal Display screens;
- Buttons and matrix keypads;
- Floor brightness sensors.

Any combination of these devices can be attached to the simulated microcontroller. Each pin of each device is modelled in the simulator as a pull up resistor (to a 5 V reference), a pull down resistor (to a ground reference), and a voltmeter. Pins are grouped into nodes, and the simulator calculates the voltage at each node based on the resistances. Microcontrollers and devices can change the voltage levels at a node by changing the resistance values at their pins, and can respond to signals from other devices by measuring the voltage levels present. Robots are defined using text files, which will allow for new robots to be easily added to the simulator.

EVALUATION OF THE SIMULATOR

In the standard Robotics Peer Mentoring course, students were asked to make the Envirocar robot:

- Flash a light;
- Drive in a zigzag pattern;
- Wait until it sees light, then charge forward;
- Follow a black line marked on a white surface.

A special version of the courseware was prepared for the simulator. The exercises were retained, but the instructions were altered slightly to describe the simulator rather than the real car. Students were given 90 minutes and asked to complete as many tasks as possible. All students were able to flash the light and make the car drive, and two of the secondary school students successfully completed all exercises. When programming the real Envirocar robot, it typically takes students between 45 and 100 minutes for the first two exercises, and between 120 and 200 minutes to complete all of the exercises. This suggests that the simulated and real robots are similarly challenging for students.

Evaluation Method

The simulator was evaluated using a user feedback. Two groups of school students were asked to use the simulator, and then complete a survey. One group of eight students came from a primary school. The group consisted of year seven students, aged approximately twelve years. All eight students used the simulator, and two completed the survey. The second group of eight students came from a secondary school. All eight of the year 10 students returned the survey.

The survey contained questions on the students' usage of the simulator, and on whether the simulator had made them consider a career in an area related to programming or robotics. For each question, the students were asked to circle one of five phrases, corresponding to strongly negative, negative, neutral, positive or strongly positive responses. The survey was kept anonymous to encourage more honest responses; although, a limitation of such arrangement is that survey responses cannot be connected to observations of the students while they were using the simulator.

Survey Results

The first three questions investigated school students' intentions about their future career plans. The students were asked if they had enjoyed programming, if they intended to study programming at a tertiary level, and if they intended to study something else robotics related. The remaining three questions asked the students about their usage of the simulator - if they enjoyed using it, if they had or would use it at school outside of robotics lessons, and if they had or were planning on using it at home.

Table 1: Survey responses.

Year Level	Enjoyed programming	Will study programming	Will study other engineering	Enjoyed simulator	Used simulator outside class	Used simulator at home
10	0	+	0	0	0	0
10	++	+	+	+	0	0
10	+	+	0	+	+	0
10	+	+	+	+	+	+
10	+	0	+	0	0	0
10	+	+	+	+	+	+
10	++	++	+	+	+	+
10	+	+	+	+	0	0
7	+	0	0	+	0	+
7	+	-	0	+	0	0

.. Legend: -- = strong negative, - = negative, 0 = neutral, + = positive, ++ = strong positive

Analysis of the Study

None of the students surveyed disliked programming. All students had prior programming experience - the primary school students using Great Cow Graphical BASIC, the secondary school students with an assembly based graphical language. The secondary school students had previously built the Envirocar robot, and created some programs for it. The students had also chosen to take part in a robotics class, indicating at least some prior interest in electronics and programming.

Only one student did not want to study programming at a tertiary level, while none were against studying other robotics related field. This suggests that at worst, using the simulator has not significantly discouraged students and that at best; they may be convinced to study engineering or IT with very little additional encouragement.

Some students were considering using the simulator outside of allocated class times, either at school or at home. This was one of the reasons for developing the simulator, and supports the work of Fagin and Merkle, who used robots to teach a computer science class and found that access to the robots outside of class was a major difficulty [10].

While using the simulator, all students appeared to be engaged in the task. On a follow up visit to the secondary school a week later, students were using the simulator by choice to test their robot programs, rather than using a physical Envirocar robot.

Some feedback regarding the user interface of the simulator was taken into account, and has led to some minor design changes since the user study. Students found the labelling of some buttons confusing, and the simulator made it difficult to control errant programs.

The primary school students were not given the opportunity to use the real Envirocar robot, as the soldering required to construct the kit was considered too dangerous for younger students. While they were still happy to use the simulator, a couple commented that they would have liked to use the real Envirocar robot as well as the simulator. If other students share this opinion, the simulator may have been more effective if used in conjunction with a limited number of real Envirocar robots. The simulator would mean that fewer real robots would be required, and so it may be possible for teachers to assemble two or three robot kits for the class to share.

The simulator has also been used outside the formal user study. It was part of a demonstration at the Project Fair Day hosted by the School of Computer and Information Science at UniSA in November 2010, and successfully simulated the Envirocar robot and other robots for several hours. The simulator was also used at a training session for the mentoring programme to illustrate how the program on a robot should behave. The working robot needed to be shown to 30 participants simultaneously, which would have been difficult with a physical robot - but by showing the simulator on a projector, this was straightforward.

Overall, the results were positive, and at this stage, it is planned for further testing and development of the simulator to be continued in following years.

FUTURE WORK

More improvements to the simulator are planned for the future. It is hoped that the software can be extended to allow students to design their own robots, and then test the design before building real robot. More robot sensors will be added to the simulator, which will allow students to play with parts that would be too expensive to use otherwise.

A development of a remote laboratory is planned in order to enable students to test the programs they have developed in the simulator on a real robot without the necessity to purchase the hardware. Some simulation features may also be added to the NetLab remote laboratory to verify the safety of student designed circuits.

It is also necessary to test the simulator on a larger group of students. Testing the simulator on a group of students who had not chosen to take part in a robotics class would provide more insight into the ability of the simulator to attract interest. This could be achieved at a school where robotics is a compulsory part of the curriculum.

CONCLUSION

Most work involving robot simulators in education has tested their effectiveness in teaching university students - students who are already enrolled in robotics related disciplines. This study has tested the simulator on pre-tertiary school students, and found that its use can be enjoyable for the students, and can encourage them to study further robotics related subjects.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the contribution of Brenton Dansie and David Chan to the project and its outcomes.

REFERENCES

1. Burhans, D., Meyer, R., VanVerth, P., Puehn, D., Steck, V. and Wiejaczka, J., Introductory computer science with robots. *Proc. 21st National Conf. on Artificial Intelligence*, Boston, Massachusetts, USA, 2, 1962-1963 (2006).
2. Nedic, Z., Dansie, B. and Nafalski, A., Effectiveness of robotics peer mentoring program for raising interest in engineering, science and technology. *Proc. Inter. Conf. on Engng. Educ. and Research - Progress Through Partnership*, Bouzov Castle, Czech Republic, 659-667 (2004).
3. Göl, Ö., Nafalski, A., Nedic, Z., McDermott, K.J. and Machotka, J., Peer tutoring in high schools to increase engineering awareness. *Proc. 7th UNESCO Inter. Centre for Engng. Educ. Annual Conf. on Engng. Educ.*, Mumbai, India, 103-106 (2004).
4. Australian Engineering Excellence Award - AusIndustry Innovation Award for Engineering Awareness Rising through CoreChart Robotic High School Mentoring. the Institution of Engineers Australia (2003).
5. Considine, H., CIS10259: GCGB Robot Simulator. Final Year Honours Project Report, School of Computer and Information Science, University of South Australia, Adelaide (2010).
6. Nafalski, A., Machotka, J. and Nedic, Z., *Collaborative Remote Laboratory NetLab for Experiments in Electrical Engineering*. In: Garcia Zubia, J. and Alves, G.R. (Eds), Using Remote Labs in Education, University of Deusto, Bilbao, Spain, 177-197 (2011).
7. Nedic, Z., Nafalski, A., Machotka, J. and Göl, Ö., *Enriching Student Learning Experiences through International Collaboration in Remote Laboratories* (2011). A book resulting from Competitive Grant CG8-697, Australian Learning and Teaching Council, Sydney, NSW, Australia, 41, 22 September 2012, http://www.olt.gov.au/project-enriching-student-learning-experience-unisa-2008
8. Corter, J.E., Nickerson, J.V., Esche, S.K. and Chassapis, C., Remote versus hands-on labs: a comparative study, *Proc. 34th Annual, Frontiers in Educ. Conf.*, Savannah, Georgia, USA, 2, F1G-17-21 (2004).
9. Nafalski, A., Nedić, Z., Machotka, J., Göl, Ö., Ferreira, J.M.M. and Gustavsson, I., Student and staff experiences with international collaboration in the remote laboratory NetLab. *Proc. 1st World Institute for Engng. and Technol. Educ. Annual Conf. on Engng. and Technol. Educ.,* Pattaya, Thailand, 40-45 (2010).
10. Fagin, B. and Merkle, L., Measuring the effectiveness of robots in teaching computer science. *ACM SIGCSE Bulletin,* 35, **1**, 307-311 (2003).