Programming for Younger Students
Linda Reynolds
Desert Robotics
teckteacher@yahoo.com

Programming for Younger Students

Introduction

As a long time Botball coach I have had many opportunities to work with groups outside of Botball. Many of the students in these situations are still in elementary school, yet they show a great enthusiasm for robotics. Some of the school programs have been with middle schools but many are elementary schools that service grades one to six or grades four to six. I also give lectures with the California State University system. Recently at the Conference for Innovation I held a day long seminar for teachers interested in robotics for their students. I was expecting mostly middle school or high school teachers, but all of my classes were predominantly elementary school teachers. They showed a great interest and wanted to be able to teach programming to their students. The regular programs that I teach are basic robotic camps, one to three days in length. So what do you do with these younger students?

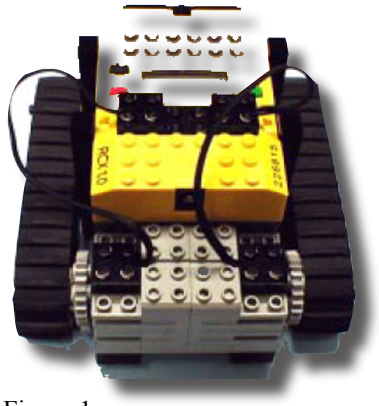

Figure 1:
Tank Bot, Small Robot with Treads

In a storage bin packed away in the back of a closet at home, I have quite a number of the old yellow RCX bricks, the kind we used to use in Botball when I first began. They don't ever seem to break; just add new double a batteries and they are ready for action. Along with Sheri Gundlach, a teacher from John Glenn Middle School, and any number of our Botball team students over the years we have put on several programs for elementary school age children using the RCX as a controller. You can build a simple robot with instructions available on the internet. [See Figure 1] This robot is called Tankbot and is built using basic LEGO pieces and has two treads for moving. [1] If you still have some of the old studded style LEGO and grey LEGO motors you can find the instructions for Tankbot on the Carnegie Mellon webstie at:

http://www.education.rec.ri.cmu.edu/roboticscurriculum/teachertraining/tankbotbldginstr.pdf

Starting with this robot you can make some nice modifications and run some exciting games and competitions. [See Figure 2]

You can load Kiss C into the RCX controller but the downloading uses IR transmitters and is not particularly easy. We have lots of RCX bricks but not many of the transmitters or cables left, so programming with them isn't practical. Instead, in our robotic camps, the students built remote controllers for their robots using the touch sensors and the remote battery boxes from LEGO. This allowed them to design and to compete in several games. [See Figure 3]

Figure 2:
Modification of Tank Bot with Wheels and a Claw

Figure 3:
Student Designed Competition Board

The students all have a great time and learn a lot about robotics but unfortunately there isn't much programming involved.

Programming With Younger Students

So how do you teach programming to younger students? Many schools use the NXT controller and the icon based programming language in elementary and middle schools. One such school is Palm Desert Charter Middle School located near me in Southern California. Reggie Clark is the Robotics Elective Class teacher and has had exceptional success with his program. The students start robotics in the 6th grade and use the NXT LEGO controller. Reggie also has several successful Botball teams and the transition from the NXT icon based language to KISS C seems to be easy for the students to make. But what about regular programming like the KISS C that is used in Botball? Is it too difficult for younger students and hence making it better to just wait until middle school or at least 6th grade? I think not.

A few years ago I had the opportunity to run a program that included what we called a 'Community Botball Team'. I was working at both an elementary school and near-by middle school running after school technology classes. My suggestion was that we offer a mentoring program where middle school students act as tutors for the younger elementary school students. Both principals from the elementary school and the middle school were receptive to the idea to create a Community Botball Team.

The team members would be sixth graders that were previously students from the elementary school that are now enrolled in the middle school. The middle school and elementary schools are on adjoining campuses so it was easy to combine after school programs. My suggestion was to pull the sixth graders back to the elementary school to become the Community Botball team. Many of our students get lost in the shuffle of the middle school, so this would provide an opportunity for them to be part of an exciting enrichment program as sixth graders. But these students would not only be Botball team members, they would also become cross-age tutors for fourth and fifth grade students.

This experiment went well and the 4th and 5th graders fit right in to the Botball teams. Some of the younger students had been in the First LEGO League program and knew the basics of robotics and others were entirely new to the program. The middle school students still made up the regular Botball team members, but all of the 4th and 5th graders worked along with the designing, building and programming of the robots. A few of the 5th graders excelled and even ended up joining the team at GCER. The key was letting each student participate in hands-on activities providing the time needed to grasp the concepts. They need to do it themselves to really understand.

The Organic Robot

As far as learning the programming, I decided to use some simplified methods for teaching KISS C. We started with a lesson on flow charts and programming logic. This enabled the students to see the necessary steps involved in programming a robot to complete a specific task. This lesson is my adaptation of Ken Boone's Organic Robot. [2] You can see his page filled with all kinds of robots and lessons at:

<http://www.kensrobots.com/>

The Organic Robot lesson was designed to be used without any robotic equipment at all; just a few props are needed. The students design a robot and develop a simple robotic programming language. Organic parts, the students themselves, are used to 'build' the robot. Ken's lesson was originally designed to be used with students in middle school, high school, and gifted elementary programs. [3] At the time I first saw his project I was teaching a technology course for 2nd, 3rd, 4th, and 5th grade GATE (Gifted and Talented Education) only classes. I modified the lesson and used it not only with the GATE classes, but also with many regular classrooms and even with a special day program. You can still download Ken's original lesson plans at:

<http://www.kensrobots.com/Organic.pdf>

Figure 4:
Giselle, Sonia, Leslie, and Mekai work on the computer

After this introduction to programming and flow charts the students started using the KISS C platform. [See Figure 4] Several of the younger students worked with the all girls team at the middle school.

Visual Aides to Programming

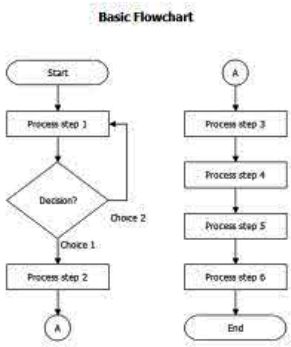

Figure 5:
Sample Flow Chart

Another lesson I designed to help the younger students understand the programming language was called Visual Aides to Programming.

The first is the flow chart. Flow charts have been around for many years. A flow chart provides a visual or symbolic representation of the steps in a process. It uses different symbols to represent actions accomplished by the process. In programming ovals, rectangles, diamonds are used to show how the program moves from step to step. Arrows designate the flow of the process. [4] [See Figure 5]

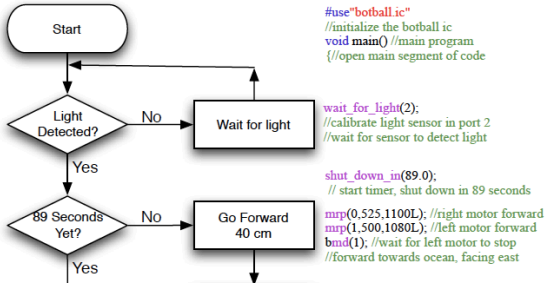

Figure 6:
Flow Chart with Code

Each symbol contains a form of pseudo code explaining what the programming should be doing at that moment. To help the students understand the **2nd step was to add the actual code** corresponding to the flow chart. The students put it right next to the

symbol so the correlation between the pseudo code and the KISS C programming language could be seen easily. [See Figure 6]

In the final step the students ran the program on the robot and took pictures of the position of the robot at different times corresponding to the code sections. The **pictures were added next to the actual code.** The code was also commented to explain what was suppose to be happening or where the robot was suppose to be.
[See Figure 7]

```
4.    mrp(0,250,200L); //right motor forward
      mrp(1,250,-200L); //left motor backward
      bmd(1); //wait for left motor to stop
      //left turn, facing north
```

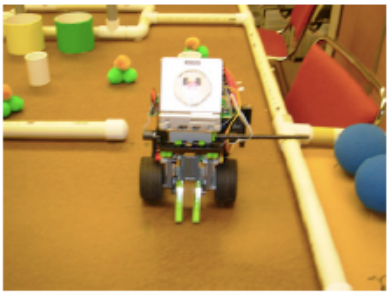

Figure 7:
Picture of Robot Position and Code

These activities proved to be very successful in teaching basic programming to younger students. The visualization of the code was very helpful and made a great project for the group.

Introduction to Robotics Class

During this June I will be working with five schools running a 3-day Introduction to Robotics class for each school. We will be using the CBC 2 controller and the KISS C software. Students will be divided into groups and each group will work as a single team building and programming a robot. The students will build the demobot from the Botball workshop and program it to complete several tasks. The class gives an overview of robotics in todays world. Using the principles of engineering, electronics, and design students will build and program a simple robot capable of completing various tasks. They will learn the basics of programming logic and syntax through use of the KISS IDE C based programming language. Through the program students will learn how to control a variety of devices such as motors, servos, sensors, and micro controllers.

Even though most of the students will be incoming 6th graders next year, I feel that they will be able to master the basic C programming needed to control their robots. I worked with a similar group during this last year. These students were familiar with the NXT icon based language but had never written any code in C. I discovered that they were very interested in the written code and were very eager to try.

Although able to program in the icon based language these students could not explain the relationship between the icons they used and what the robot was doing; they just knew which

icon to use. We explained how to write a simple function as a substitute for the icon to move the servo on their robot. This approach worked quite well and the students seemed to understand how the code worked. Telling them that each icon was actually a section of code that a professional programmer had written and now they were going to be a programmer and write their own code, was very motivating. Even though their code was simple they managed to write a function to control the servo on their robot . Then we added to this simple code the ability for the servo to move at different speeds. They were very pleased with themselves when they finally got it to work correctly.

Major Goals of the Class:

- Develop an understanding of the relationship between science, technology and engineering.
- Demonstrate an ability to solve problems and think critically by completing a challenging project.
- Learn the process of design, project management, teamwork and strategic thinking.
- Understand the development of flowcharts and the key elements of programming.

Conclusion

Programming with younger students takes some extra time and patience, but I feel is well worth the effort. These students can be the next generation of Botballers and future designers, programmers, and engineers.

The main steps to being successful with younger programmers is

1. Explain each step of the process thoroughly
2. Give adequate time for the students to practice and understand the steps
3. Make everything hands-on and real-world; no abstract ideas
4. Encourage the KISS approach (Keep It Simple)

If you do this then programming with younger students will be a great project with many rewards.

References:

[1] Tankbot Building Instructions
<http://www.education.rec.ri.cmu.edu/roboticscurriculum/teachertraining/tankbotbldginstr.pdf>

[2] Welcome to Ken Boone's Robotic Home Page
<http://www.kensrobots.com/>

[3] The Organic Robot
<http://www.kensrobots.com/Organic.pdf>

[4] What Is a Flow Chart?
<http://www.breezetree.com/articles/what-is-a-flow-chart.htm>