PEEK 2003 Fourth Grade Curriculum

Jen Lewellen & Roz Echols

General Objectives:

- Introduce the RCX as a component of a Lego system
- Introduce the concept of outputs from the RCX (to motors and lights) and inputs (from light sensors, touch sensors, etc)
- Introduce RoboLab as a tool
    - Use RoboLab programming to develop graphs
    - Use RoboLab to perform a single task multiple ways
    - Introduce the concept of computer programming (how it works, what can be done, etc.)
- Discuss concepts of graphing, calibration, and measurement as necessary companions to an engineering project
- Discuss how light and touch sensors work
- Team competitions
    - Use concepts developed throughout semester to achieve goals
    - Smaller competitions: building a car that moves, graphing results, using gears and sensors in cars
    - Major competitions:
        - Stop at the line 3 different ways
        - Navigate through a maze using calibration and sensors

Lesson 1: Lego Construction Basics

Lesson Objective: To familiarize the students with the RCX and how it can be incorporated into a Lego system.

Learning Objective: To learn about RCX concepts such as the difference between inputs and outputs, what components attach to what parts of the RCX (motors, sensors, etc), and how the system has to be connected as a whole.

The Challenge: During this lesson, the students will simply be experimenting with the RCX and its many uses. The final product for the day will not be too spectacular, but the students should have a good grasp of the possibilities.

Materials:

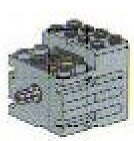
- Lego Simple Machine kits or other Lego building pieces

Vocabulary:

Procedure: Start off the lesson by discussing the concept of an RCX - why you need it, how it is powered (batteries), and any other interesting features. Then, start to discuss the specifics of input and output:

OUTPUTS

Explain that the RCX has three outputs labeled A, B, and C. Using wires, we can hook up motors and lights to these outputs and integrate them into our designs.



Take one motor, one wire connector, and the RCX from your kit. Connect one black connector end to port A on your RCX and match the other end to the black square on the motor. Press the ON-Off button on the RCX. Press the program button (labeled PRGM) on the RCX several times and notice the number changing on the display from 1 - 5. The number on the RCX indicates the active program. Keep pressing the program button until it displays the number 1 and press the green run button. What is happening now? Press the RUN button again and the motor should stop.

Now, disconnect the wire from the RCX, turn it 180°, and reconnect it. Press the RUN button again. What do you notice that is different?

Take the second motor and wire from your kit. Connect this second motor to port C using the connector, as you did with the first motor. Press the RUN button again. Now what is happening? Disconnect the wire from the RCX, rotate it 180°, and reconnect it. Press the RUN button again. What do you notice that is different?

Now you can see that it is possible to have multiple outputs operating at once. These motors can all be operating in various directions and at various speeds at the same time.

INPUTS

Explain to students that the RCX has three inputs labeled 1, 2, and 3. We can connect sensors to these ports since your Lego kit comes stocked with two touch sensors and a light sensor. The touch sensors have a yellow button sensor that connects to the RCX with the same connectors used with motors. The light sensor is a blue brick that has a built-in connector permanently attached. The RCX can sense the light intensity picked up by the light sensor and measure it on a scale of one to one hundred.

Light Sensor - Connect the light sensor to port 2 on the RCX. Press the VIEW button on the RCX several times. Notice the arrow that moves around the RCX screen. Keep pressing the view button until the arrow points at port 2. Notice the number that is on the screen. This is the light level that the sensor is picking up on a scale of one to one hundred.

Notice the light that is on the front of the light sensor. Aim this light at the space where you are working. Hold the sensor about six inches off the surface of the desk with the light pointed down. Now, move the sensor down while keeping your eyes on the reading on the RCX screen. What happens as the sensor gets closer to the table? Why?

Touch Sensor: Connect the two touch sensors to inputs 1 and 3. Connect the two motors to outputs A and C. Press the PRGM button until program 2 is selected. Press the green RUN button. The two motors should start to rotate. Press the touch sensor connected to port 1, then the touch sensor connected to port 3. What happens to the motors? Press the RUN button again. Disconnect the touch sensors and reconnect them in different orientations. Press the RUN button once more. Has anything changed?

Extensions:

Assessment: After the lesson, ask students the following questions to refresh their memory:

- How would you add a motor into a series of gears? What Legos from your kit would you use?
- How would you mount a motor to the RCX?
- If you were building a car, how many motors would you need?
- If you wanted twice as much power and would therefore need two motors, how would you construct the gearing to do so?

- If a light sensor is pointed at a light bulb, what value will the RCX display?
- If the RCX can understand 100 different light values, how many different values does the RCX understand for the touch sensor? What do you think these values are?
- What will happen if you hook up a touch sensor to port B (i.e. the difference between an input and an output)

Troubleshooting:

Resources:

- Lego/Tufts website- www.ceeo.tufts.edu/curriculum
- Lego Dacta "Simple and Motorized Machines" Teacher Guide

Lesson 2: Building a car that moves

Lesson Objective: To familiarize students with techniques of building a sturdy car using gears, motors, and the RCX.

Learning Objective: To learn construction techniques (how to build a solid car), gears (torque, teeth, sizes, etc), and how to properly connect pieces.

The Challenge: To have each team construct a car that incorporates the RCX, gears, and the motors, and is fairly well-built (i.e. doesn't fall apart every time it is picked up).

Materials:
- Lego Simple Machine kits or other Lego building pieces

Vocabulary:

Procedure: Begin the lesson with some introductory questions:
- What are gears? What are their important features?
- How do we use multiple gears together?
- Why are there different sizes of gears(Here it is good to give a brief intro to the concept of torque - not the math and physics description, but a basic qualitative explanation that torque is a force acting at a distance. Doors are great examples. Have a kid try to close a door by pushing near the hinge, and then by pushing farther out. Students can also practice with a friend by holding their arms out, trying to push at different points, and seeing how easy or difficult it is to resist)
- How do we use gears for our cars? (transmission, fast/slow, power vs. speed)

Engineering Challenge/Programming: The challenge of the day is for students to use gears to construct two cars - one that goes as fast as possible (without doing any programming) and one that can go up an incline.

Extensions:

Assessment: After the lesson, discuss with students how they used gears to make their cars go fast. How about to climb a hill?

Troubleshooting:

Resources:
- Lego/Tufts website- www.ceeo.tufts.edu/curriculum
- Lego Dacta "Simple and Motorized Machines" Teacher Guide

Lesson 3: Intro to RoboLab and Graphs

Lesson Objective: To give the students a very basic idea of computer programming and RoboLab.

Learning Objective: To learn how to use RoboLab to program cars.

The Challenge: To use Pilot 1 to program the cars (from last time) to run for a certain length of time, recording the distance each time. Using this data, students will develop graphs and thus, be able make their cars run for a specified distance.

Materials:
- Lego Simple Machine kits or other Lego building pieces
- Graph paper
- 'Go the Distance!' worksheet

Vocabulary:

Procedure: Before beginning the challenge, discuss the following questions:
- What is RoboLab? How do you use it? What do the different symbols mean?
- How do you load a program into your car? (proper technique, etc.)
- How will the students be using RoboLab in this activity?

After this discussion, begin the engineering challenge. Students should have cars they built the week before, and thus should be ready to experiment with RoboLab and get right into graphing. (Check with teacher beforehand to make sure students understand basic graphing.)

Engineering Challenge/Programming: Today's challenge is to build and program a previously-built car to drive a specific challenge distance. At the end of the lesson, all groups will compete to see whose car stops closest to the challenge distance.

Extensions:

Assessment: Ask students the following questions:
- How well did your car perform? How could you improve it?
- If you wanted to make your car run for a much longer distance, how would you figure this time out from your graph? How about for a shorter distance?
- How does the computer program from RoboLab change your car?
- Do you getter better results if your car goes fast or slow?

Troubleshooting:

Resources:
- Lego/Tufts website- www.ceeo.tufts.edu/curriculum

- Lego Dacta “Simple and Motorized Machines” Teacher Guide

Go The Distance!

Challenge: To build and program a car to drive a specific challenge distance. At the end of the lesson, your group will compete to see whose car stops closest to a challenge distance.

Group name:

Group members:

Jobs: (alternate jobs with each new program time)
• **Programmer** - writes and loads program into car.
• **Operator** - carries car to test area and runs the tests.
• **Recorder** - measures and enters the distance the car drives into the test chart.

1. Fill out the test chart below.

TIME	TEST #1	TEST #2	TEST #3
1 second			

2. Create a line graph plotting time against distance using the information from your test chart. Use graph paper.

3. When given the challenge distance, use your line graph to make your best time estimate. Program your car to drive your estimated time, but <u>do not test!</u>

Lesson 4: Supersized Going the Distance

Lesson Objective: To combine skills learned about the two types of sensors with calibration skills learned in last week's lesson.

Learning Objective: To learn that it is possible to solve the same problem multiple ways, and that it is important to choose a method that works best for the particular situation.

The Challenge: To program cars to go a set distance using timed calibration, light sensors, and touch sensors.

Materials:
- Lego Simple Machine kits or other Lego building pieces

Vocabulary:

Procedure: At the beginning of the lesson, remind students what they learned about inputs from the light and touch sensors (recap how these work if necessary).

Next, demonstrate how to use RoboLab to program with sensors (using the more advanced Pilot levels). Discuss how the sensors can be used to make the car perform specific tasks.

Engineering Challenge/Programming: Instruct students to program their cars to go exactly 1m (or 1 yard) using the timed calibration, a light sensor (a dark line on the ground), and a touch sensor (an obstacle at the end).

Extensions:

Assessment: At the end of the lesson, ask students the following questions:
- Which method worked the best for this activity?
- How did the car know when to stop for each method?
- How did the RoboLab program differ for each method?

Troubleshooting:

Resources:
- Lego/Tufts website- www.ceeo.tufts.edu/curriculum
- Lego Dacta "Simple and Motorized Machines" Teacher Guide

Lesson 5: Navigate a Maze

Lesson Objective: To tie together all of the skills students have learned thus far into one final competition

Learning Objective: To reiterate all concepts learned this year and to allow students to practice using them all together.

The Challenge: To use already-built cars to navigate a maze by using light sensors and tough sensors, or simply calibrating their way through the maze.

Materials:
- A maze (Discuss with the teacher beforehand whether he/she wants to design the maze, or whether you should do it.)
- Lego Simple Machine kits or other Lego building pieces

Vocabulary:

Procedure: Begin this lesson with a 10-minute discussion reviewing a line graph from Lesson 3 and discussing the various parts and vocabulary of the graph. This lesson reinforces Lesson 3 and also introduces multiple-step Pilot 4 programming, which allows students to program turns into their car. (The teacher may change the units of distance measurement from inches to centimeters to initiate a discussion of standard versus metric measurements.)

After the preliminary discussion, students should begin the engineering challenge.

Engineering Challenge/Programming: The challenge of the day is for students to program their cars to navigate a maze constructed by the teacher. Students should do the following three steps in the given order:

1. Preliminary Investigation: Calibration of Car

   Student groups should program cars (pre-built around a RCX programmable brick) to run for several designated times: 1 second, 2 seconds, and 3 seconds. They should observe the distances, in metric units, that their cars travel and develop a basic line graph plotting time against distance.

   (Note: At this time, the teacher should tape out a challenge maze consisting of four 90° turns.)

2. Developing a Hypothesis

   In order to write the four-step RoboLab program, students must measure and chart the distance between each turn of the maze. Students should estimate from their graphs how much time they need to program their cars to run for in each of the three steps. (These estimations should be based on the distances students previously measured.) After deciding on times, students should enter this information into their four step RoboLab programs. They must also determine and enter the direction of each turn. (Note: The angle of the turn is determined by how long the car turns. The time value for the turn is variable and can either be preset by the teacher or calibrated by the students. This feature of Robolab could also augment a lesson about angles.

3. Testing and Adjusting the Hypothesis

   Students should download their new programs onto their cars and run them. Instruct them to note their accuracy in maneuvering the maze and adjust their time estimates accordingly.

Extensions:

Assessment: After allowing each group's car to navigate the maze, hold a group discussion about the problems and solutions the different groups encountered. The class can also discuss how changing the units of distance from inches to centimeters affected their graphs. You can even investigate, or follow-up, the relationship between inches and centimeters by examining the ratios of the different distances graphed.

In addition, ask students the following questions:

- How did you use each of the different techniques to navigate the maze?
- Which was the easiest? The hardest?
- How did you use RoboLab for this project?
- Was it easier to navigate the maze with a fast car or a slow car?

Troubleshooting:

Resources:

- Lego/Tufts website- www.ceeo.tufts.edu/curriculum
- Lego Dacta "Simple and Motorized Machines" Teacher Guide