Continuing in Engineering
How we modify our robots for an obstacle course

I. Overview

Brief Description: Build a sturdy car that is compact, and which turns well enough to navigate an obstacle course

Subject: Engineering, technology

Topics: Trouble shooting, building sturdy cars, designing for speed, mechanics of turning, using gears and motors

Grade 5 Accommodations for SPED: Visual pop-quiz, diagrams, charts

Lego Materials: RCX, batteries, RoboLab sets, Lego motors

Additional Materials: Visual diagram of motors, gears, pulleys; Design Notebooks; large charts to record time trials; markers

Content Standards: Math: (tbc) Science/Technology: Standard 9, 10, 11. Strand 3, BPS: Recognize design need or engineering problem; Develop, sketch and discuss possible solutions and select one; analyze a product's components and their functions; Use, evaluate, suggest ways to improve object, explore, analyze structure, propulsion, control guidance and suspension systems in land, air, water and space vehicles

Literature: Galimoto (story of a young inventor in Africa) (tbc)

Writing: (tbc)

II. Discussion

Obstacle Course Considerations: Discuss: 10 minutes

Show the obstacle course. Why does your vehicle need to be sturdy? What other changes are needed? Draw analogy to one’s body: basketball player crouching before turning, figure skaters spinning, etc. Show student examples of compact vs. ‘non-compact’ vehicles

Discuss sensors (touch switches) and operation of Program 2.

Lynne’s suggestions: (10 minutes) Discuss some new ideas which have been drawn onto a poster, ask the students if they can figure how they would work:

a. show a motor and two gears and a motor with two pulleys: discuss how to change speeds with these.

b. show how a gear assembly (rack & pinion) can help a vehicle turn corners - show a real model and/or draw the model and discuss

Quick Rules of Thumb:
Robot not nimble enough? Smaller wheels; adjust gearing or pulleys to slow it down; make it sturdier; use a shorter wheelbase; lower to ground; swiveling wheel(s) or glide(s) for front

III. In-Class Activity

Before students arrive:

Check out obstacle course design. Prepare/draw visual (diagram) of gears, pulleys, etc. Make copies of handout: “Rules of Thumb: Checklist for a Sturdy Car”. Make sure firmware is downloaded to RCX units and that program 2 is functional.
**Implementation:**
Students have kits of previously built robots. Students provided rubric of tests robots need to pass: time and speed, obstacle course, rough surface (?), straight line (?), forward and reverse (?), ...
Perform time tests of robots on obstacle course (using last week's speed design); each group does two time trials: 10 minutes
Students rebuild their robots using new information: Make them sturdier and better able to handle the task at hand.
Perform time tests of rebuilt robots on obstacle course; each group does two time trials: 10 minutes
Debriefing: What did we learn? What went right? Wrong? 5 minutes
Total: 35 minutes _____________________

**Extension:**
Prepare an instruction book complete with illustrations: "How to Build A Fast Robot " or “How to Build a Sturdy and Maneuverable Robot” (for Trotter Science Expo)
Moving into Programming (3/20/2002)
How we teach our robots how to behave

I. Overview

**Brief Description:** Introduce students to programming using RoboLab Pilot and Inventor.

**Subject:** Engineering, technology

**Topics:** formal thinking, computer programming

**Grade 5 Accommodations for SPED:** Color code diagrams/ICONS, use concept maps or charts to show or diagram ideas, use ‘icon charts’ for reference notebooks, use visual references (print out important concepts of slide show/laptop slides), present lesson as concept rather than segmented

**Content Standards:**
- Math: (tbc)
- Science/Technology: Standard 9, 10, 11. Strand 3, BPS: Recognize design need or engineering problem; Develop, sketch and discuss possible solutions and select one; analyze a product's components and their functions; Use, evaluate, suggest ways to improve object, explore, analyze structure, propulsion, control guidance and suspension systems in land, air, water and space vehicles.

**Literature:** Galimoto (story of a young inventor in Africa) (tbc)

**Writing:** (tbc)

II. Discussion

**Introductions:**
- Upcoming BPS Robotics Competition: 2 min.: Angel and Carrie
- Housekeeping discussion: Kits and missing components

  “To ensure that you are able to make a working robot to enter the competition, you need to have all of your pieces; teams will be responsible for keeping all pieces together in your kits. Nothing will be added to or replaced: we cannot make any more substitutions.”

  Suggestions for keeping kits together:
  1. Check under tables before passing in kits;
  2. One person makes sure that your kit stays on your side of the table;
  3. If your robot breaks apart, collect all the pieces - some were outside in C-Pod!!

- Transition: 2 minutes: Lynne

  Introduce students to the idea of having their vehicles perform smarter (more flexibly), faster, more intelligently.

  Review things we need to pay attention to for sturdy, fast cars: motors, alignment, batteries, wheel size, attaching beams or plates to hold parts together, etc.

  What about using robots to do many things: smarter, faster, better? Programming!

**Discussion of General Programming ideas:**
- important elements of the Peanut Butter and Jelly sandwich lesson
- how to write a program
- introduction to smart robots smart cars and smart robots need:
  - directions that tell them how to do things so they can do these things on their own, and they need to do them carefully and in an orderly manner as when telling
someone how to make a peanut butter and jelly sandwich.

A few simple things we can program our robots to do; we can tell them:

- how fast to go;
- whether to go forward or backwards;
- how to stop and start;
- how to do something over and over again;
- how to do something a certain number of times;
- how to avoid dark places or light places;
- how to navigate tricky areas, do loops, go around sharp angular corners
- what to do if they bump into things

**Discussion of RoboLab Pilot and Inventor (Angel/Carrie):**

"Our cars were hard to control. Not exact - did not do exactly what we wanted. Programming allows us to be more precise."

Demonstrate RoboLab PILOT (levels 1-4 if possible) using the projector and laptop from TechBoston: 15 minutes

Pass out Fred’s icons sheet. Discuss how pictures (icons) carry directions for computer to read and execute. Discuss sequence of icons.

Pass out RoboChallenge sheet

### III. In-Class Activity

**Before Class:**

- **Lego Materials:** RCX units, fresh batteries (AA & 9V), RoboLab sets, esp. Lego motors, Pilot & Inventor CD, serial/USB tower, RoboLab 2.0 or 2.5 loaded on all machines, firmware downloaded onto RCX units;
- **Additional materials:** laptop computer & projector (RoboLab demo), Design Notebooks
- **Other:** Make copies of icon handout: “How to Program”, and “RoboChallenge #1” sheet

**Implementation:**
- Students have kits and previously built robots already on tables
- Students program their robots using RoboChallenge sheet - adults circulate and help
- Student use their Design Notebooks to record or draw any modifications.
- Debriefing: What worked and did not work?
Continuing in Programming
How we teach our robots how to behave

Note: The way we’ve worked so far has consisted of a lot of segmenting. How about a change in course? Present a problem (challenge) to introduce a new concept, in this case, programming. Questions we might pose:

- How can you create a robot that behaves the way you want? One that does something, and does it faster, smarter, sings it, etc.?
- Or more specifically, how would you program your robot for an Olympic event?

I. Overview

Problem: How can you make your robot behave the way you want it to, and how can you modify it so to go faster, be smarter, even ‘think’ its way around angles, corners, and against time limits?

Brief Description: Present a large programming challenge; Olympic preparation

Subject: Engineering, technology

Topics: Formal thinking, computer programming, trouble shooting, building sturdy structures, designing for speed and/or maneuverability, introduction of mechanics of gears and motors

Grade 5 Accommodations for SPED: Color code diagrams/ICONS, use concept maps or charts to show or diagram ideas, use ‘icon charts’ for reference notebooks, use visual references (print out important concepts of slide show/laptop slides), present lesson as concept rather than segmented

Content Standards: Math: (tbc) Science/Technology: Standard 9, 10, 11. Strand 3, BPS: Recognize design need or engineering problem; Develop, sketch and discuss possible solutions and select one; analyze a product's components and their functions; Use, evaluate, suggest ways to improve object, explore, analyze structure, propulsion, control guidance and suspension systems in land, air, water and space vehicles.

Literature: Galimoto (story of a young inventor in Africa) (tbc)

Writing: (tbc)

II. Discussion

Introduction: how we will extend our programming and take on more difficult challenges

Discussion: review last class, hand out new sheets

III. In-Class Activity

Before Class:

Lego Materials: RCX units, fresh batteries (AA & 9V), RoboLab sets, esp. Lego motors, Pilot & Inventor CD, serial/USB tower, RoboLab 2.0 or 2.5 loaded on all machines, firmware downloaded onto RCX units;

Additional materials: laptop computer & projector (RoboLab demo), Design Notebooks

Other: Make copies of icon handout: “How to Program”, and “RoboChallenge #2” sheet

Implementation:

- Before students arrive: have kits and previously built robots already on tables
- Students program their robots using RoboChallenge sheet #2 - adults circulate and help
- Student use their Design Notebooks to record or draw any modifications.
- Debriefing: What worked and did not work?

**Challenges:**
Today’s Engineering Challenge: Go around an obstacle course using the touch sensors faster than any other car; make the fastest straight line road racer in the class

Today’s Programming Challenge: Choose one or more programming challenges form the handout

<Can we Project student programs on wall ? Try to have discussion using projector rather than on rug.>

**Summary:** Teaching your robot to behave means

- Part 1: Deciding what to teach them
- Part 2: Learning to program in an icon-based programming environment such as RoboLab including, as needed, how to use the computers
- Part 3: Pilot levels 1, 2, 3 and sample programs of Robots that behaved
- Part 4: Extreme Behavior: Using Inventor and sample programs

**IV. Additional Materials**

**Design Notebook:** The Design Notebook An important part of Art, Science, Engineering, and especially Robotics is being able to keep track of what you did so that you can:

- share your work with other people
- go back to work on problems that you couldn't solve earlier, but that you've got the resources or abilities to do now
- pick up where you left off after an interruption

A useful tool for this is a notebook, and is among the most valuable things you will take with you from this experience. It will be a record of your work, and will be something you refer back to often. The notebook should be personal. That is, it should reflect you: your tastes, your words, your thoughts. It will contain recipes for building things, ideas for things you'd like to build, feelings about your robots and this experience - even poetry, if you wish. You should take time after every exploration to record notes about what you did. Use a combination of words, quick little drawings or sketches (called "thumbnails") that you make; you can even include pictures and clippings printed out from the Web, books, magazines, etc. I favor using pencil and very few colors.

Try to address each of these after every class:

- the initial problem - what you did to solve it
- what materials and other resources you used
- what might you change for next time?
- a recipe with enough detail so you could do it again

Here’s an example of mine:

<paste in example>