Unit 6:

Robotics

Introduction

Robotics provides a physical application of the programming and problem solving skills acquired in the previous units. The LEGO® Mindstorms NXT software uses drag and drop programming which will provide a natural transition from Scratch. Robots are shared by several students which will emphasize the collaborative nature of computing. In order to design, build and improve their robots, students will need to apply effective team practices and understand the different roles that are important for success.

Discussing the features of robots provides an opportunity to emphasize how computing has far-reaching effects on society and has led to significant innovation. Students can discuss such topics as:

- The effects innovations in robotics have had on people.
- The significance of processes that have been automated because of robots.
- How innovations in robotics have spurred additional innovations.

The unit consists of three main sections:

- The features of robots (Days 1-3)
- Familiarization with the robot and the software (Days 4-13)
- Robotics projects (Days 14-33)

Throughout the unit the similarities and differences between Scratch and the programming needed to move the robot can be highlighted.

Specific topics for each instructional day are listed in the overview chart on the next page.

Daily Overview Chart	
Instructional Day	**Topic**
1	What is a robot? Identify the criteria that make an item a robot.
2-3	Evaluate robot body designs and create algorithms to control robot behavior.
4	Set up LEGO® Mindstorms® NXT® kit.
5	Build robot base.
6-7	Introduce the features of NXT Brick—the “brain” of the robot.
8-9	Introduce the features of the Mindstorms NXT software.
10-13	Program the robot using the Mindstorm Robot Educator Software tutorials.
14	Introduce RoboCup real life robotic competition and write instructions for tic-tac-toe.
15	RoboTic-Tac-Toe Tournament and introduction to RoboCupJunior Dance Challenge.
16-18	Build, program, and present a dancing robot.
19-23	Build program and present a rescue robot.
24-33	Final projects and presentations

Daily Lesson Plans

Instructional Day: 1

Topic Description: "What is a Robot"? Identify the criteria that make an item a robot.

Objectives:

Students will be able to:

- List and explain the criteria that describe a robot.
- Determine if something is a robot, using the criteria.

Outline of the Lesson:

- Brainstorm about robot definition (10 minutes)
- "Kismet" video (5 minutes)
- Elements of a robot (10 minutes)
- Am I a Robot? Activity 1 (15 minutes)
- Student group work—Are we Robots? (15 minutes)

Student Activities:

- Brainstorm what they think of when they hear "robot" and then identify common features of robots.
- Participate in whole class activity determining if common items are robots.
- Work in small groups to complete "Are we Robots?" activity.

Teaching/Learning Strategies:

- Brainstorm: Ask students what they think of when they hear "robot". Display responses. Responses may include the following:
    - Movie and TV robots such as Wall-E, iRobot, Robots, Rosie from The Jetsons
    - Modern industrial robots such as those involved in assembly-line factory work
    - Mars Rovers
    - iRobot robots, both the vacuum cleaner and the robots built for military use, other robots such as bomb detection and detonation
- View the video "Kismet" from Teachers Domain.
- Ask students if they can identify common features of the robots they have identified. What do all those robots have in common? What tasks are easy for robots? What tasks are hard for robots? (Answers: robots are often used for dangerous or repetitive tasks such as recovering bombs, search and rescue operations in dangerous conditions where the robots search and the humans rescue, factory work. They are replaceable, unlike humans, and don't get bored or make

mistakes when doing the same thing over and over. Tasks that require judgment or human-like interaction such as recognizing when there is a problem or walking and seeing like humans are hard for robots. The two articles listed in the resource section provide more information and would be interesting for students to read.)

- Use the What is a Robot? handout to guide a discussion of robots.
- Hand out copies of Am I a Robot? activity, with the pictures of a basic stove and a fancy microwave. Check with students to make sure they recognize the items in the two pictures. Based on student input, display the five criteria for whether something is a robot: body, input, program, output, behavior. Note that what distinguishes a robot from a programmable device is the ability to respond to changes in the environment and adapt; robots respond to. Explain to the class that as a group you will figure out whether each of the two machines shown is a robot.
    - Go through the stove first. Ask students to figure out whether the stove meets the criteria for a robot:
      Body—yes
      Input—yes (dials to turn the burners off and on, set oven temp)
      Programmable—yes, in the sense that oven temperature tells a sensor what temperature the oven needs to be heated.
        - Remind students that they programmed in Scratch and that the programmable aspect of the robot will require a language to provide the robot with instructions

      Output—yes (heat!)
      Behavior—yes, the oven responds by stopping at the desired temperature. It also adapts to changes as in opening the oven door, adding a frozen item, etc. by adding more heat to get back to the desired temperature.

      Next go through the microwave in a similar way:
      Body—yes
      Input—yes (buttons)
      Programmable—yes (buttons set time, set mode, microwave can be programmed by the user, for example “cook 3 minutes 50% power, hold 1 minute, cook 1 minute 90% power)
      Output—yes (microwaves in chamber, light comes on)
      Behavior—yes (cooks food, makes popcorn, boils water…)
      Question: Does a microwave adapt?
- Hand out copies of Are we Robots? activity two. Explain the directions. Either have students brainstorm machines as a group to complete the table or have them think of machines on their own. Have students work in small groups to complete the table, determining whether each machine is a robot according to the criteria.
- Optional Extra Credit—have students research Isaac Asimov’s three Laws of Robotics. What are the three laws? What is law Zero? Why did he come up with these laws and how do they think

these laws affect our thinking about robots today?
Law Zero: A robot may not injure humanity, or through inaction, allow humanity to come to harm.
Law One: A robot may not injure a human being, or through inaction, allow a human being to come to harm.
Law Two: A robot must obey the orders given to it by human beings, except where such orders conflict with Law One.
Law Three: A robot must protect its own existence, as long as such protection does not conflict with Laws One and Two.

Resources:

- Jennifer Casper and Robin Murphy, Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center, IEEE Transactions on Systems, Man and Cybernetics 33:3, 2003, pp. 367-85.
- Robin Murphy, J. Kravitz, S. Stover and R. Shoureshi, Mobile robots in mine rescue and recovery, IEEE Robotics & Automation Magazine 16:2, June 2009, pp. 91-2003.
- What is a Robot? Handout (Based on handouts from The Big Picture "Robotics Teacher Guide 1" (Item #29852 from LEGO Dacta))
- Am I a Robot? Activity
- Are we Robots? Activity (Based on handouts from The Big Picture, "Robotics Teacher Guide 1" (Item #29852 from LEGO Dacta))
- http://www.teachersdomain.org/resources/eng06/sci/engin/design/lp_robot/index.html specifically http://www.teachersdomain.org/resources/eng06/sci/engin/design/kismet/index.html (may require free registration)
- Asimov's three laws of robotics: http://en.wikipedia.org/wiki/Three_Laws_of_Robotics, http://www.asimovonline.com/asimov_FAQ.html#series13 , essay at http://www.sfwriter.com/rmasilaw.htm

What is a Robot? Handout

There are many different kinds of robots, from ones designed to build cars to ones that vacuum to ones that explore other planets. To be a robot, a machine must meet certain criteria. A machine is only a robot if it has all the elements listed below:

Body

The body is a physical substance and shape of some type. The body will be designed based on the function—some look like vehicles, some like an arm, and some like a person. If you can touch it, that's the body.

Control

Control is a program to control the robot. Robots must be told what to do. To control a robot we need:

Input

Input is the information that comes from the robot's sensors. Robots have sensors that they use to get information from the robot's environment. For example, a smoke detector can detect smoke. (In other words, sensing the robot's environment). Robots typically have external and internal sensors.

Programmable

The program is a set of instructions or rules that the programmer gives the robot. For example, a smoke detector has a program to make a sound if it senses smoke. To be a robot, a machine must be programmable.

Output

The output is the action a robot takes, often involving motors, lights, or sounds. For example, a smoke detector makes a loud sound and might flash lights. (In other words, effecting change in the robot's environment—adapting.)

Behavior

Behavior is the combination of outputs that result in the task or job the robot does. For example, the behavior of a smoke detector is to "go off" in the presence of smoke. "Going off" is a combination of making noise and flashing lights, and may also involve calling the fire department.

Am I a Robot? Activity

Image 1: Basic Stove

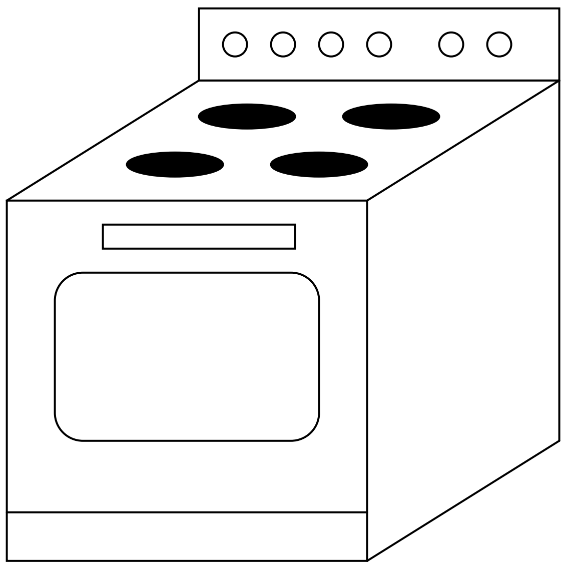
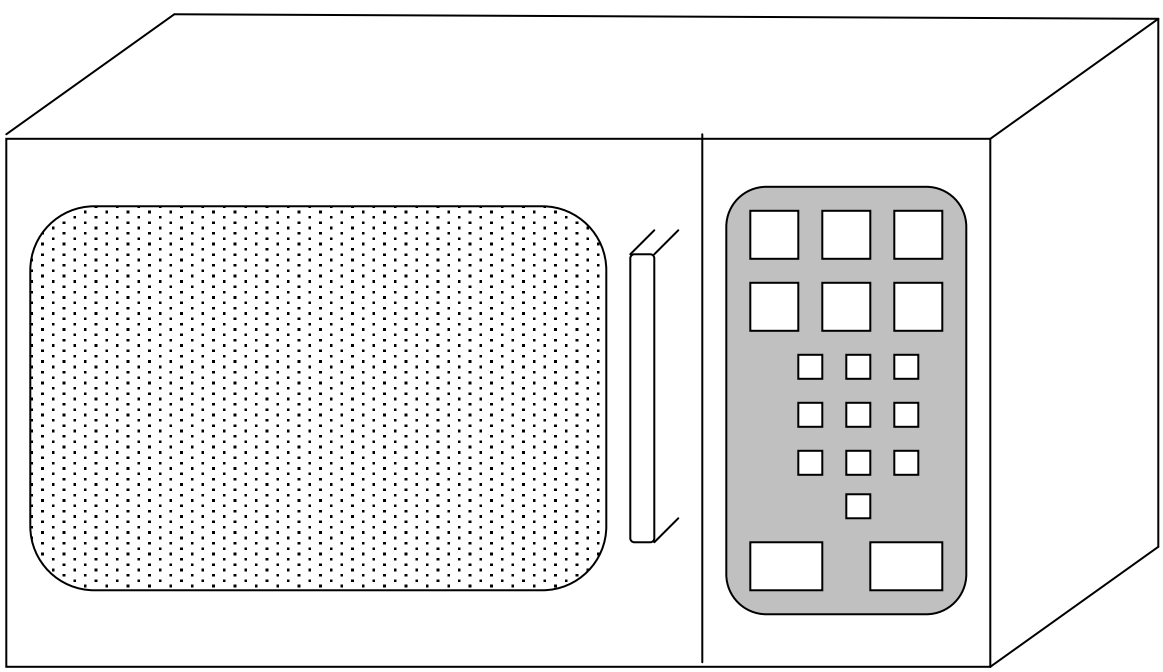


Image 2: New Microwave



Are we Robots? Activity

Instructions: Below is a list of machines that you may encounter in your daily life. Add machines to the bottom. Complete the table by deciding if the machine meets the criteria for being a robot. Then determine if the machine is a robot.

Body—physical form of some kind
Control—
  Input—gets information from sensors, buttons, etc.
  Program—Is programmable, follows a set of instructions you give it
  Output—an action it takes
Behavior—what it does; the function it performs

	Body	Input	Program	Output	Behavior	Is it a robot?
Stove						
Microwave						
Radio						
iPod						
Flashlight						
Bicycle						
Car						
Alarm clock						
Traffic light						
Photocopier						
Computer						
Mars Rover						

Instructional Days: 2-3

Topic Description: Evaluate robot body designs and create algorithms to control robot behavior.

Objectives:

Students will be able to:

- Evaluate how the design of a robot's body affects its behavior.
- Create an algorithm to direct a human "robot" from one part of the room to another.

Outline of the Lesson:

- "Are we Robots" activity (15 minutes)
- Journal Entry (5 minutes)
- The effect of changing design (15 minutes)
- Student group work—Can a robot tie your shoes? (40 minutes)
- Student group work—Walk like a robot (35 minutes)

Student Activities:

- Participate in discussion of "Are we robots" activity.
- Complete journal entry.
- Discuss how changing the design of an item affects the item.
- Students work in pairs to try tying a shoe in several robot-esque situations including with closed eyes, with tongue depressors, pliers, and with another person.
- Students work in small groups to direct a person to move along a path given a limited list of commands.

Teaching/Learning Strategies:

- Revisit "Are we robots" activity. Go through the list of items, asking students to indicate if they thought each item was a robot or not. Occasionally, especially if there is disagreement, ask students to defend their answer.
- Journal Entry: What happens when you change the design of a robot?
    - Have students share their responses
    - Ask students, "If you could change the body of the printer (or another device in the room) what would you change? How would that affect other things like the behavior or function of the printer, price, cost to build, or popularity? Have students share their ideas.
- Explain that there are limits to what robots can do because robots are limited by their bodies. For example, it is difficult to create a robotic hand that can grasp small or delicate items—it

would require many motors (simulating all the muscles in the hand) and many sensors to detect the item (simulating the neurons in the hand).

- Make sure each pair of students has a shoe that can be tied.
- Direct students to first try tying the shoe blindfolded or with eyes shut. Discuss how it went—Was it hard? What was hard about it? How was it like a robot tying the shoe?
- Direct students to tie the shoe with heavy gloves on. Discuss the experience. How was it like a robot tying the shoe? What made it hard?
- Direct students to tie the shoe with tongue depressors taped onto thumbs and forefingers or just holding tongue depressors. Discuss the experience. How was it like a robot tying the shoe? What made it hard?
- Direct students to tie the shoe with pliers. How was it like a robot tying the shoe? What made it hard?
- Direct the students to work with their partner to tie the shoes using the pliers, each person holding one pair. Discuss the experience. How was it like two robots working together? What made it hard?

- Activity: Walk like a robot
    - Choose one student to be a "robot" or tell students that you will be the robot. Choose a starting point and an ending point between which the "robot" must navigate. Make sure the path is not direct.
    - Tell the class that they must direct the robot from the starting point to the ending point using only five commands:
        - Turn left 90 deg.
        - Turn right 90 deg.
        - Take a step forward with the left foot.
        - Take a step forward with the right foot.
        - Stop.
    - Students can take turns or work as a group. The robot should only follow those five commands and not respond to other commands. Tell students to be careful with the robot and not walk it into walls or barriers. (The robot should stop before it hits a barrier such as a wall.).
    - At some point, remind students about loops. They can tell the robot to repeat a command or a block of commands such as "repeat: take a step forward with the left foot, take a step forward with the right foot until you are at the wall"
    - Point out that this is frequently what is done in dancing and choreography—sequences of steps are repeated.
    - If there is time, show the video of the "macarena" referenced in the resource section.
    - Conclude by pointing out that these kinds of commands are what they will be programming their robots to execute.

Resources:

- Activity: Can a robot tie your shoes? (From www.thetech.org/robotics/activities/page05.html)
- Materials: shoes that tie, tongue depressors, masking tape, heavy gloves, pairs of pliers, blind folds
- Walk like a robot activity from LEGO Materials.
- http://www.cs.colorado.edu/~lizb/chaotic-dance/macarena-orig.mpeg.gz
- Explanation of video: http://www.cs.colorado.edu/~lizb/chaotic-dance.html

Instructional Day: 4

Topic Description: Set up LEGO® trays.

Objectives:

Students will be able to

- Distinguish between the LEGO parts for building a robot.

Outline of the Lesson:

- Distribution of LEGO kits (10 minutes)
- Separation of LEGO parts into the appropriate compartments of the trays (45 minutes)

Student Activities:

- Student groups work together to set up their LEGO kits for use in building robots.

Teaching/Learning Strategies:

- Give each pair (or group of three) a LEGO® Mindstorms® NXT® kit. Point out the picture that shows where each item should be placed in the tray.
- Ask students to set up their trays so that they will be ready for use in building robots.

Resources:

- LEGO Mindstorms NXT kit

Instructional Day: 5

Topic Description: Build the base of the robot.

Objectives:

Students will be able to

- Assemble the base of the robot.

Outline of the Lesson:

- Explanation of LEGO Mindstorms manual (10 minutes)
- Assembly of base of robot (45 minutes)

Student Activities:

- Student groups assemble the base of the robot.

Teaching/Learning Strategies:

- Have students get out their kits and the manual that comes with the kit. Go through step 1 on p. 8 with the students to make sure they understand the format of the manual.
- Ask student groups to assemble the base of their robot according to the instructions on pp. 8-21. (Batteries should be charged in advance.)

Resources:

- LEGO Mindstorms manual

Instructional Days: 6-7

Topic Description: Introduce the features of the NXT Brick—the "brain" of the robot.

Objectives:

Students will be able to

- Distinguish between the parts of the NXT brick.
- Hook up input and output devices correctly.
- Use built-in NXT Brick programs.

Outline of the Lesson:

- Observation of the NXT brick (20 minutes)
- Plug in sensors, motors, and light, and run "View" programs (30 minutes)
- "Try Me" built in programs (40 minutes)
- NXT Brick programs (20 minutes)

Student Activities:

- Articulate what they observe about the the NXT brick while the teacher explains each part.
- Test sensor data using the 'View' programs and report observations.
- Run 'try me' programs and describe what the programs do.

Teaching/Learning Strategies:

- Have students get out their robot base, sensors, lights, motors, and three wheels. Explain that the NXT is the brain of the robot. Have students describe the parts they see and make sure the following parts get identified. (The NXT User Guide pp. 9-12 can be used as support.)
  - Ports number 1-4: these are input ports. You use wires to plug sensors into the NXT brick. There are four kinds of sensors: touch sensors (detect touch/obstacles), sound sensor (detect the sound levels), light sensor (detects light level), ultrasonic sensor (detects movement and distance to an object). Reminder: input means sensing something in the robot's environment.
  - Ports A-C: these are output ports. You use wires to connect devices for output. The devices are lamps and motors. Also, note that the speaker is an output port. The output is that the light can go on or off and that the motor can turn or stop turning. Reminder: output means effecting change in the robot's environment.
  - Buttons:

    Orange button: On/Enter

    Light grey arrows: Navigation, left and right

Dark Grey button: Clear/Go back. Keep pressing this to turn off until prompt, and then hit orange

- Lines in the right side: speaker. This is where noise comes out of the robot.
- If the rechargeable battery is in, there will be a power plug and LCD lights.

- Tell students to turn on the robot by pressing the orange button. What happens? (It makes a happy little song. LOUD.) What do they see now?
  - NXT at the top—name of the brick. This can be changed in the software
  - Battery level top right.
  - Running icon—next to the battery icon. As long as it is spinning, the NXT is turned on and working correctly. If it freezes, the NXT has frozen and must be reset.
  - There are three icons on the screen. The one that is highlighted by default looks like two floppy disks and has the label above "My Files". If they start hitting buttons, they can scroll through several menu options by using the arrow keys or go into My Files by hitting the orange button. The menu options are:
    - My Files—where programs will be stored.
    - NXT Program—allows you to build small programs using only the NXT without the need for a computer.
    - View—you can do a quick test of your sensors and motors and see the current data for each. You have to select the test you want to do and which port the sensor or motor is on. Only one test can be run at a time. The data will display on the screen.
    - Bluetooth—you can set a wireless connection between the NXT and other Bluetooth devices including other NXTs, phones, and computers.
    - Settings—you can change settings such as the speaker volume and the sleep time.
    - Try Me—built-in programs.
  - Explain that in order for the robot to really do anything, you have to hook up input and output devices. Ask students to try to identify the devices in the kit. Make sure they can identify the touch sensor, sound sensor, light sensor, ultrasonic sensor, servo motor and lamps. Reinforce that the sensors are all input devices and the motors and lamps are output devices.
- Demonstrate and have students carefully plug the devices into the NXT. Sensors can be plugged into any input port numbered 1-4 but these default settings are used for the test and sample programs. See pp. 5-6 and 9 of the NXT User Guide for more information. Make sure students know to support the weight of the devices and the NXT brick without pulling on the wires.
    - Port 1: Touch sensor
    - Port 2: Sound Sensor
    - Port 3: Light Sensor
    - Port 4: Ultrasonic Sensor
    - Port A: Light

        - Port B & C: Servo motor
- Have students navigate to the View menu. They should test each of the sensors and see what the displays do. Make sure they also use the motor rotations and motor degrees program. See NXT User Guide pp. 23-33 for more information. After a few minutes with students experimenting, ask what they noticed. What kind of data does each of the sensors provide? How could a robot use this in a program? Remind them that what they are doing is testing and debugging.
- Have students navigate to the Try Me menu by using the dark gray button to move up the menus and using the light gray arrows and orange button to enter the Try Me menu.
    - Select one of the programs and have all the students try it. Once they have tested it, ask them what it did. See if they can flowchart what the program does.
        - Try sound—moves the motors faster as more sound is detected.
        - Try touch—changes display and makes noise when button is touched.
        - Try light—makes noise based on how much light is detected.
        - Try ultrasonic—changes noise based on distance detected.
        - Try motor—changes sound based on motion of motor on port A.
- Finally, have students navigate to the Program menu and follow the directions in the LEGO Mindstorms manual on pp. 22-45, trying the programs indicated. They should then test the programs and make sure each one works as expected. (Also see the NXT User guide pp. 15-16 for more information.)

Resources:

- NXT User Guide

Instructional Days: 8-9

Topic Description: Introduce the features of the Mindstorms NXT Software.

Objectives:

The students will be able to

- Recognize the parts of the Mindstorms NXT software.
- Explain the different types of icons in the common palette and how to use them.
- Explain the different types of icons in the complete palette and how to use them.
- Explain the difference between software errors and hardware errors.
- Explain the difference between logical errors and syntax errors.

Outline of the Lesson:

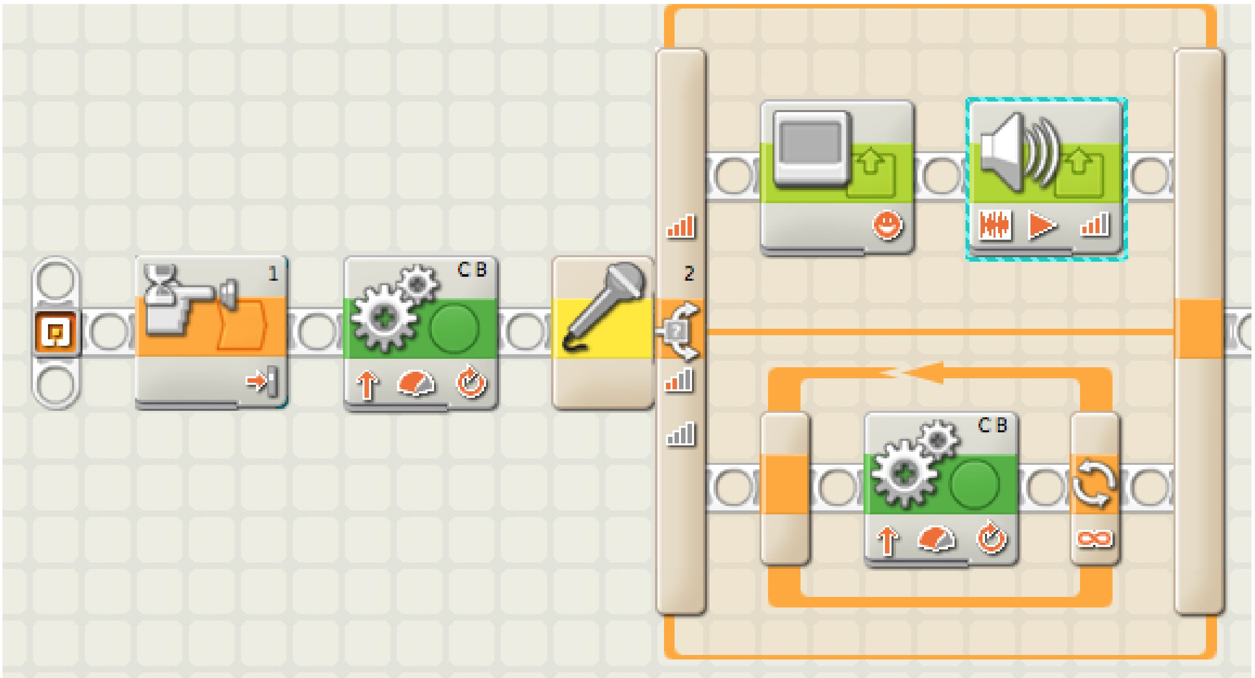
- Review of Program activity from Day 7 (20 minutes)
- Interface: the parts of the Mindstorms NXT software (10 minutes)
- A simple program from the common palette (30 minutes)
- A simple program from the complete palette (40 minutes)
- How to use the tutorials (10 minutes)

Student Activities:

- Discuss how the programs were created in the NXT brick and how they behaved compared to expectations.
- Listen to explanation of Mindstorms NXT software and respond to questions.
- Give ideas to teacher as s/he writes small programs in the software.
- Listen to explanation of how to use the tutorials.

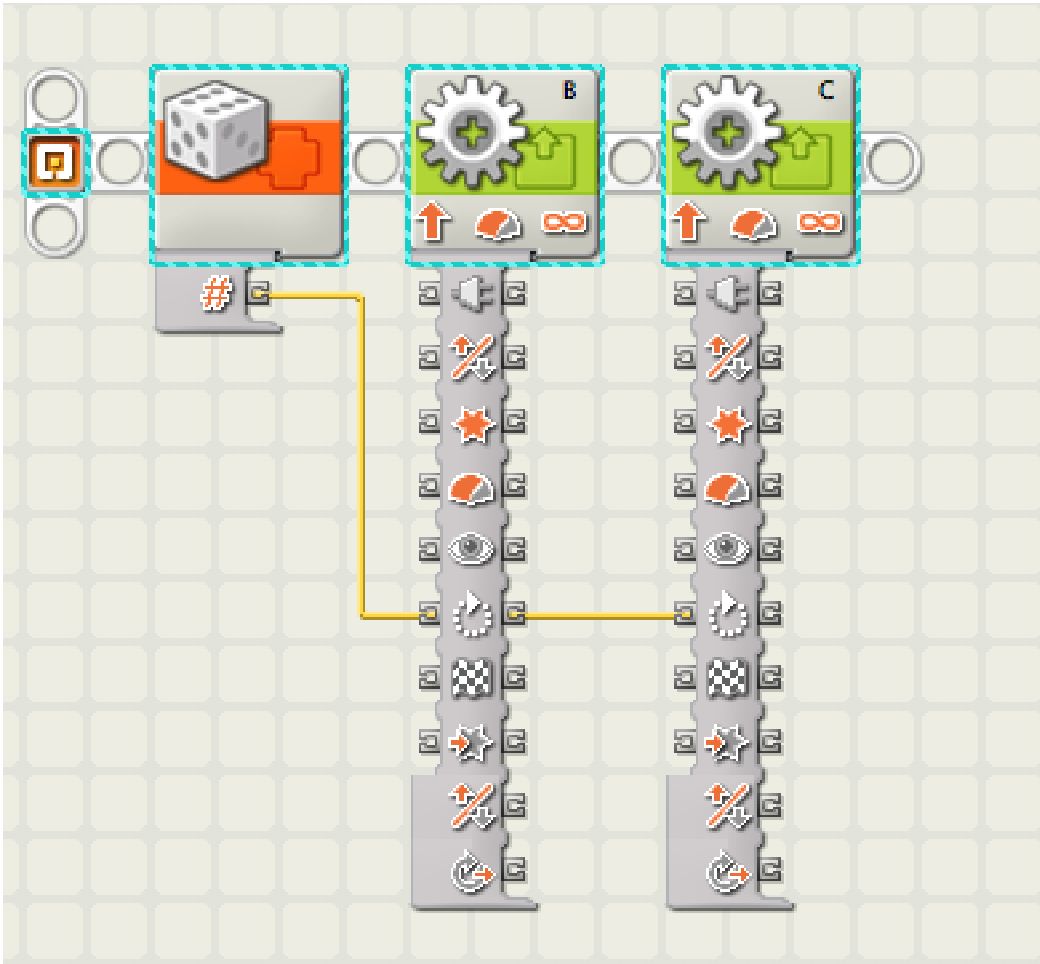
Teaching/Learning Strategies:

- Ask students what they programmed the robot to do. Get several answers. Did it do what they expected? Why or why not? Would it be a good idea to use the NXT Program interface to write all their programs? Why not? (It can only take 5 commands in a program.)
- Projecting the teacher's screen, launch the Mindstorms NXT software. Show the students where the tutorials are in the Robot Educator section and how to open a new program. Using the User Guide pp. 48-49, describe all the parts of the interface.
- With student input, use the common palette to build a small program. Ideally, use a variety of the blocks of the common palette, explaining what each one does as you use it. For example, if you wanted to build a program that told the robot to wait until the touch sensor was touched, then move forward for one rotation then listen and if a loud sound occurs, then display a smiley face and play a sound otherwise move forward, it would look like this:



- Save the program and download it to an NXT brick. Make sure the brick is set up to do the actions—have one built with the driving base and any necessary sensors. Demonstrate the running of the program.
- Modify the program and download it again. Try to make mistakes during this period and show how to debug the program by frequently testing it, downloading extra blocks, and also making mistakes such as having disconnected blocks. During this part have students try to work with the software themselves and follow along with you.
- Open a new program and switch to the complete palette. Show the differences in the two palettes. With student input, write a new program using the blocks of the complete palette. Show the differences in controlling the program. Make sure to show how to wire things in the data hub. For example, a program that runs the motors for a random amount of time would

look like this:



- Make sure to make mistakes and demonstrate how to solve problems with the software such as mis-wiring ports. Have students try these features at their seats as you do it. Point out the similarities between programming the NXT software and what they did in the last unit with Scratch.
- This is a good point at which to discuss
    - Software vs. hardware errors—in robotics the programming may be correct, but the robot configured incorrectly.
    - Syntax vs. logical errors—the program may compile, but the logic can still be incorrect.
- Tell students that the next five days will be spent going through the tutorials in order to learn how to build and program the NXT system.

Resources:

- NXT Robot Educator

Instructional Days: 10-13

Topic Description: Program the robot using the Mindstorm Robot Educator Software tutorials.

Objectives:

The students will be able to:

- Use the building blocks of the common palette to program the robot.
- Build robots that can execute the functions programmed through the Robot Educator Software.
- Program the robot using some or all of the complete palette of blocks.

Outline of the Lesson:

- Description of the assessment model (10 minutes)
- How to use the tutorials (10 minutes)
- Build and program robots according to tutorials (255 minutes)

Student Activities:

- In groups of 2-4, students follow tutorials to build and program small robots.

Teaching/Learning Strategies:

- Explain assessment model for tutorials. (Recommended: observe some but not all robots, such as those for tutorials 8, 16, and 20 in the common palette along with several from the complete palette; look at robot construction and the program as well as execution to determine grade.)
- All students should complete the tutorials for the common palette before moving on to the complete palette. It will be helpful for the future projects if students complete most, if not all, of the tutorials for the complete palette as well.
- Circulate throughout class to answer questions, help troubleshoot, and assess robots.
- If some groups finish early, have them assist other groups.

Resources:

- NXT User guide pp. 50-53 explain the tutorials

Instructional Day: 14

Topic Description: Introduce RoboCup real life robotic competition and write instructions for tic-tac-toe.

Objectives:

Students will be able to

- Explain how a sequence of game moves can be expressed in simple statements.
- Describe the RoboCup challenge and examine how robots have been programmed to play soccer.
- Develop if-then statements and use Boolean operators to direct a human "robot" to play tic-tac-toe.

Outline of the Lesson:

- Tic-tac-toe (10 minutes)
- "Robot Competitors Meet on a Soccer Field of Dreams" (25 minutes)
- Instructions for a "robot" to play tic-tac-toe. (20 minutes)

Student Activities:

- In pairs, students play a game of tic-tac-toe; then they discuss and write answers to the posted questions.
- Read and discuss the article, "Robot Competitors Meet on a Soccer Field of Dreams".
- In pairs, students write a series of clear instructions for a "robot" to play tic-tac-toe.

Teaching/Learning Strategies:

- Before students enter the classroom, write the following on the board or chart paper: "Play a game of tic-tac-toe with your partner. Then think about these questions together, and write your answers: What are the rules of tic-tac-toe? What decisions does a player need to make before taking a turn? How would you verbally describe each of these decisions? What is the action a robot would need to take based on the decisions?"
- After a few minutes, have students share some of their responses. Make a list of the rules of tic-tac-toe on the board. Ensure students remember that if statements and conditionals are required to describe the moves of the game. Collect the written responses to the warm up activity.
- Distribute the article "Robot Competitors Meet on a Soccer Field of Dreams" and have students read it.
- Lead a discussion about the article.

- Explain to students that they will be working in pairs to write an application for human "robots" (students will act as the robots) to enable them to play tic-tac-toe. The following day will be the RoboTicTacToe Challenge. Remind them of the earlier discussion of tic-tac-toe. What goals does each player have? Who starts the game? Is there a "best place" to put the first X? What are some winning strategies for the next move? For example, If the X is in the center, then where should an O be placed? Why is "if-then" logic a good way to explain strategy for a simple game like tic-tac-toe? How can Boolean operators, and/or/not, help simplify the commands?
- Demonstrate the opening move for a game of tic-tac-toe on the board. Draw a nine-space grid and label the squares one through nine. Then ask students where to place the first X. Depending on where it is placed, have students create an if-then statement that determines the next move. For example, "If the first X is in the center, place an O in a corner square."
- Ask students to complete the instructions. Each instruction in the entire sequence will cover every possible combination of moves the students can think of until a game is completed. Students need to remember that there are multiple options for each move (including the beginning move). They should consider all of the possibilities in developing their code. They also need to consider what the behavior the robot will exhibit based on the instructions provided.
- Note that the focus in this lesson is really a reinforcement of programming as a set of instructions in the context of something most students understand. The game of tic-tac-toe is not a natural example of robotics because robot environments are generally dynamic with infinite possible states of the environment.

Resources:

- Lesson plan from NY Times Lesson Plan Archive: http://www.nytimes.com/learning/teachers/lessons/20010802thursday.html
- Copy of article http://www.nytimes.com/learning/teachers/featured_articles/20010802thursday.html
- Dictionary

Instructional Day: 15

Topic Description: RoboTic-Tac-Toe Challenge and Introduction to RoboCupJunior Dance Challenge.

Objectives:

Students will be able to

- Debug conditional statements by testing them and compete as teams in a RoboTic-Tac-Toe Challenge.
- Describe dancing robots that have competed in the RoboCupJunior Dance Challenge.

Outline of the Lesson:

- Debugging of robotic-tac-toe statements (5 minutes)
- RoboTic-Tac-Toe challenge (35 minutes)
- Introduction to RoboCupJunior Dance challenge (15 minutes)

Student Activities:

- Complete debugging tic-tac-toe statements by testing that they work correctly in several games.
- Participate in RoboTic-Tac-Toe challenge.
- Listen to an explanation of RoboCupJunior Dance Challenge and watch videos of dancing robots from RoboCupJunior challenges.

Teaching/Learning Strategies:

- Ask students to quickly test their tic-tac-toe instructions to make sure they are complete and correct. They should play tic-tac-toe following only the instructions they have written.
- Explain the challenge: each team will be acting as a single robot "programmed" by the application they developed. One student will read a command from their application and the other student will execute the command. Teams play against each other, testing how successful their code is. Each game should be observed by the rest of the class and monitored to ensure the teams only execute the commands read.
- At the conclusion of the challenge, celebrate the winning team. Ask the students to describe why that team won? What have they learned? How would they improve their programs? (Remind students that precise instructions are required in programming.)
- Explain that RoboCup is a research initiative founded in 1997 by an international group of scientists interested in defining a common problem that could be addressed by researchers in robotics, engineering, and artificial intelligence. Most participants are university and industry research labs. RoboCupJunior (RCJ) was founded in 2000, with a focus on education. The RCJ Rescue challenge was piloted in 2001 and adapted in 2003. RCJ is open to students up to age 19. There are two divisions: primary, which is up to age 14, and secondary, which is age 14 to

19. The first two robot projects will be based on the RoboCupJunior program. The first one is the dancing robot which is the introductory level of the RoboCupJunior program. Students will build and program a robot that dances. Show videos of dancing robots in competition.

Resources:

- RoboCupJunior videos: http://rcj.robocup.org/videos.html
- More videos available through YouTube such as http://www.youtube.com/watch?v=25sZr3u-WwU

Instructional Day: 16-18

Topic Description: Build, program, and present a dancing robot.

Objectives:

The students will be able to:

- Use the NXT and output devices to build and program a robot that dances in time to music.

Outline of the Lesson:

- Explanation of project guidelines and show dance floor (15 minutes)
- Design, build, and program dancing robot (150 minutes)
- Dance challenge (30 minutes)
- Reflection and Clean up (25 minutes)

Student Activities:

- Agree on ideas and music for robot.
- Build robot.
- Write a program in Robot Educator software.
- Test robot and refine program and hardware.
- Participate in dance challenge and discussion.
- Complete project reflection. Take robots apart and put materials away.

Teaching/Learning Strategies:

- Hand out requirements and rubric. Explain guidelines and answer questions.
    - A dance floor can be made out of large square of one or more pieces of butcher paper.
- Circulate and make sure students are on task; answer questions as needed.
- Before the dance challenge, assign one student as timekeeper and another as DJ. Collect each group's program as they compete and immediately assess the robot using the rubric, while the next group gets set up. You may declare a winner or have the students vote for the best robot.
- Discuss the various dance routines and the features of each. Have students provide comments.
- At the end of the challenge, have each student complete the project reflection and submit it, then clean up the robots.

Resources:

- Official RoboCupJunior Dance Challenge rules (2008): http://rcj.robocup.org/dance.html
- Dancing Robot Activity
- Dancing Robot Sample Rubric

- Project Reflection

Dancing Robot Activity

The dancing robot assignment is based on the first level of RoboCupJunior, an international competition. More information about RoboCupJunior is available at http://rcj.robocup.org.

Task:

Build a robot that dances to music for 1-2 minutes.

Requirements:

- The robot should not take any input, only have output in the form of various dance moves.
- Dance must be 1-2 minutes long. You have a total of 5 minutes to get set up, have the robot dance, and get out of the way for the next group.
- The robot must stay in the marked space.
- The robot must be autonomous. Other than hitting the start button, no human can touch it while it performs.
- The dance should be choreographed to the music you provide. The music must be appropriate for playing at school—no obscenities, etc.
- Teams may restart the robot up to 2 times at the discretion of the teacher. Any re-started, unless due to a problem not the fault of the team, will result in a grade penalty.
- Teams are encouraged to be as creative and entertaining as possible! Props, costumes, and varied dance moves are encouraged. You may dance alongside your robot.
- Each team must print out its program and hand it in at the same time that they compete.
- Fair play is an important part of the RoboCup competition. Teams are expected to help other teams as needed and not deliberately interfere with or damage other teams' work. All students are expected to respectfully watch all other teams compete.

Process:

1. Brainstorm ideas about how your robot should look, how it should work (wheels? Arms?) and how you'll build it. Select music.
2. Start building your robot.
3. Build a program that directs the robot to do your dance moves.
4. Test and revise the program. Make sure it runs for 1-2 minutes. Make sure it matches the music. Make sure it won't fall apart!
5. Show off the robot during the dance in class.

You will have two class periods to build and program the robot, and then you will present it on the third day.

Performance will be judged on

- Programming (e.g.: use of loops, jumps, conditionals, etc)

- Choreography (e.g.: robots to move in time with music, and change actions as music changes tempo or rhythm. Choreography of humans and robots will be scored separately, etc)
- Construction (i.e., robots should be of sound construction, components should not fall off , appropriate use of gearing, smooth and reliable operation, interesting movements, effective use of mechanics to achieve a purpose, etc.)
- Entertainment Value (i.e., How much does the performance entertain or delight the audience? Originality and creativity of the presentation, etc.)
- Costume (Costume of humans and robots will be scored separately.)
- Cooperation between teams

Dance stage will be a flat area. Official RobocupJunior stage size is 6X4 m.

Dancing Robot Sample Rubric

	Extra Credit	A	B	C	F
Programming	Program uses advanced techniques including blocks from the complete palette, flow blocks, etc.	Program is straightforward and efficient, using loops and parallel sequences as necessary. Program directs attached output devices to dance.	Program is straightforward and easy to understand. Program is inefficient and could use constructs such as loops.	Program is poorly written or difficult to understand. Program has unused parts or does not correctly control robot.	Program does not work.
Choreography	Dance has at least 10 different dance moves. Dance matched music precisely. Robot changed actions as music changed tempo or rhythm	Dance has at least 6 different dance moves. Dance is varied and entertaining. Dance is choreographed to match music	Dance has at least 4 different dance moves. Dance is repetitive. Dance lasted for 45-60 seconds or 120-150 seconds.	Dance has 3 different dance moves. Dance lasted for 30-45 seconds or 150-210 seconds. Dance did not match music.	Robot did not move or did not appear to dance.
Construction	Robot constructed using advanced gearing or other advanced construction techniques. Robot demonstrates extraordinary creativity.	Robot is of sound construction: nothing falls off, robot works as intended. Mechanics used well to achieve dance moves desired.	Robot dances as intended, but some extraneous parts fall off.	Robot does not work as intended, but does move. Robot falls apart. Very simple construction – mechanics not used well.	Robot falls apart or does not move at all. Construction appears careless or haphazard.
Entertainment Value	Presentation is unusually creative. Humans dance with robot. Costume, props, etc enhance robot.	Audience is entertained by robot, presentation, etc. Robot runs correctly the first time.	Presentation is not smooth: robot must be restarted.	Problems occur but robot does eventually run mostly correctly.	Robot does not compete.
Cooperation	Student(s) helped other groups	Student worked well with group. Student participated actively in all parts of project.	Student worked somewhat well with group. Student participated in most parts of project.	Student had trouble working with group. Student participated in few parts of project.	Student did not participate in project. Student sabotaged others' work.

Robot Project Reflection

For each member of your group, evaluate their performance as a team member:

Name: Circle one word to describe his/her performance

_____________________________ Excellent Good Average Poor

Why?

____

_____________________________ Excellent Good Average Poor

Why?

____

____________________________ Excellent Good Average Poor

Why?

____

What was your favorite thing about this project?

____

____

____

If you could do this project over, what would you do differently?

____

____

____

____

Instructional Days: 19-23

Topic Description: Build, program and present a rescue robot.

Objectives:

Students will be able to:

- Build and program a robot that uses input and output devices to count simulated people by following a black line and counting “people” on the path.

Outline of the Lesson:

- Explanation of project guidelines and floor (15 minutes)
- Design, build, program robot (195 minutes)
- Rescue Robot challenge (50 minutes)
- Reflection and clean up (15 minutes)

Student Activities:

- Brainstorm how to build and program the robot.
- Build the robot.
- Write a program in Robot Educator software.
- Test the robot frequently and refine program and hardware.
- Participate in rescue challenge.
- Complete project reflection. Take robots apart and put materials away.

Teaching/Learning Strategies:

- Hand out requirements and rubric. Explain guidelines and answer questions. Show students the arena with the victims laid out. Explain that they must use sensors so that the robot will follow the black line and will sense when it has encountered a victim or a gap.
- Circulate the room and make sure students are on task; answer questions as needed.
- During the rescue challenge, assign one student as timekeeper and one to keep track of victims found. Collect each group’s program as they compete and immediately assess the robot using the rubric, while the next group gets set up.
- At the end of the challenge, have each student complete the project reflection and submit it, then clean up the robots.

Resources:

- Rescue Robot Activity
- Rescue Robot Sample Rubric

- Project Reflection
- Official RoboCup Jr Rescue Competition Rules (2008): http://rcj.robocup.org/rescue.html
- Instructions for building modules are available at http://rcj.sci.brooklyn.cuny.edu/rcj2010/rescue_suggestedbuildinginstructions.pdf . Alternatively use white butcher paper on the floor with black electrical tape as a path. Use green electrical tape to indicate victims.

Rescue Robot Activity

The rescue robot assignment is based on the second level of RoboCupJunior, an international competition. More information about RoboCupJunior is available at http://rcj.robocup.org. This robot simulates robots sent to rescue people during natural disasters. It must find "victims" along the path through each "room" and avoid obstacles. The goal is to program a robot that uses sensors to respond to different stimuli.

Task:

Build a robot that follows a black line on a white background, counts green or metallic "people" and avoids obstacles.

Requirements:

- The robot must follow the black line and attempt to complete the course through the entire arena. The robot will begin at the starting location in the doorway of the first "room".
- The robot should stop and flash a light for at least two seconds to indicate it has found a victim. For extra credit, count the number of victims and display the count.
- The robot should be able to avoid items of debris blocking the black line.
- If a robot has been stuck or lost the black line for more than 20 seconds, the teacher may pick it up and put it back onto the black line a little beyond where it ran into problems. The 20-second rule allows it to try to find its way back to the line without intervention. A team may decide to quit if the robot is faulty or repeatedly loses the line.
- Robots must be controlled autonomously except for being started by a member of the team.
- The robot will have 10 minutes to complete the course and identify all victims.
- Each team must print out its program and hand it in at the same time that they compete.
- Fair play is an important part of the RoboCup challenge. Teams are expected to help other teams as needed and not deliberately interfere with or damage other teams' work. All students are expected to respectfully watch all other teams compete.

Process:

6. Brainstorm ideas about how your robot should work: what sensors will you need? What motors and lights? What programming constructs will you need?
7. Start building your robot.
8. Build a program that controls the robot
9. Test frequently and revise the program. Make sure it correctly detects victims and that it can follow the line. Check if it can navigate gaps.

You will have three and a half class periods to build and program the robot; then you will present it in class.

Official Rules available http://rcj.robocup.org/rescue.html

Official RoboCupJunior Rescue Challenge

5.1. Victims:

5.1.1. Ten (10) points are awarded for each victim located by the robot. The robot indicates that it has found a victim by stopping and flashing a lamp for at least two (2) seconds.

5.1.2. Extra points are NOT awarded for the same victim being located more than once.

5.2. Gaps in the black line:

5.2.1. Ten (10) points are awarded for each gap in the black line that the robot successfully negotiates (i.e. recovers the line on the far side of the gap).

5.3. Debris blocking the black line:

5.3.1. Ten (10) points are awarded for each item of debris blocking the black line that the robot successfully avoids (i.e. moves around the debris and recovers the line).

5.4. Rooms:

5.4.1. Ten (10) points are awarded for each room that the robot navigates successfully (i.e. enters through one doorway and exits through the other doorway).

5.5. Ramp:

5.5.1. Thirty (30) points are awarded for the robot successfully negotiating a ramp without any assistance.

5.6. Penalties:

5.6.1. Two (2) points are deducted for each false victim identification (i.e. whenever a robot indicates that it has found a victim at a location where there isn't one).

5.6.2. Five (5) points are deducted for each lack of progress (i.e. whenever human intervention is required to enable a robot to resume progress along the black line).

Official Rules available

http://rcj.robocup.org/rcj2008/china-rescue-rules-page.pdf (Note: This references the RoboCupJunior 2008 Rescue rules. The committee members were Ashley Green, Maverick Luk, Eli Kolberg and Bill Freitas. You may wish to work with the most up to date version.)

Rescue Robot Rubric

	Extra Credit	A	B	C	F
Victims	Found victims are counted and count is displayed	All victims correctly identified	Most victims correctly identified	Some victims correctly identified	No victims correctly identified
Gaps		All gaps navigated correctly	Most gaps navigated correctly	Some gaps navigated correctly	No gaps navigated correctly
Debris		Robot avoided all debris	Robot avoided most debris	Robot avoided some debris	Robot unable to avoid debris
Rooms		Robot entered all rooms through one door and exited through the other	Robot entered most rooms through one door and exited through the other	Robot entered one room and was unable to exit	Robot did not enter the first room
Construction	Robot constructed using advanced gearing or other advanced construction techniques. Robot demonstrates extraordinary creativity.	Robot is of sound construction: nothing falls off, robot works as intended.	Parts of robot fall off. Very simple construction – mechanics not used well.	Robot does not work as intended, but does move. Robot falls apart. Robot is unable to navigate due to construction	Robot falls apart or does not move at all. Construction appears careless or haphazard.
Programming	Program uses advanced techniques including blocks from the complete palette, flow blocks, etc.	Program is straightforward and efficient, using loops and parallel sequences as necessary. Program uses sensors and strong logic to navigate challenges and find victims.	Program is straightforward and easy to understand. Program uses inefficient logic to navigate challenges and find victims.	Program is poorly written or difficult to understand. Program has unused parts or does not correctly control robot. Program does not correctly use sensors to control motion.	Program does not work.
Cooperation	Student(s) helped other groups. Managed own role & helped group members.	Student worked well with group. Student participated actively in all parts of project.	Student worked somewhat well with group. Student participated in most parts of project.	Student had trouble working with group. Student participated in few parts of project.	Student did not participate in project. Student sabotaged others' work. Made it difficult for group to work.

Instructional Days: 24-33

Topic Description: Complete Design Challenge final project.

Objectives:

Students will be able to:

- Design, build, and program a robot that solves a stated problem.

Outline of the Lesson:

- Explanation of project guidelines (15 minutes)
- Distribution of challenges (10 minutes)
- Design, build, and program robot (~7.5 class periods)
- Design challenge gallery walk and discussion (1 class period)
- Clean up (1 class period)

Student Activities:

- In groups, determine who will complete each of the four roles.
- Use the planning document to plan the robot.
- Design, build, program, and refine a robot which meets the challenge.
- Set up their robot and participate in a gallery walk.
- Disassemble the robots and carefully organize all the robotics equipment.

Teaching/Learning Strategies:

- Hand out requirements, planning document, and rubric. Explain guidelines and answer questions.
- Hand out challenges. Allow students to trade challenges as necessary. You may choose to have each group working on a different challenge or have them overlap.
- Approve planning documents as students finish plan and prepare to build and program robot.
- Circulate and make sure students are on task; answer questions as needed. At the end of each day, remind information specialists to fill out paperwork and remind groups to clean up the space. Optionally, have students fill out the daily group evaluation.
- During the design challenge, fill out each rubric as you observe the robot. If possible, videotape (or have a volunteer videotape) the running of each robot. Discuss the features of the various robots and designs.
- On the final day of the unit have students disassemble the robots and organize the equipment.

Resources:

- Design Challenge Sample Rubric
- Information Specialist Report
- Project-Reflection
- Daily Group Evaluation
- Challenges:
  - Option 1: Challenges from Design Challenges for computer-controlled LEGO products by Len Litowitz. (Litowitz-challenges.doc) Some of these challenges are more appropriate than others.
  - Option 2: Gary Stager's LEGO Challenges available from http://www.stager.org/LEGO/challenges.pdf (stager-challenges.pdf) Not all of these challenges are appropriate.
  - Option 3: Webquest

Final Project

Design Challenge Planning

STEP #1 TASK DEFINITION

Determine the purpose of your challenge—What are we supposed to do?

Criteria—list the specifications the robot needs to meet

1.

2.

3.

4.

5.

STEP #2 TASK BREAK-DOWN

List the steps the robot will need to go through to accomplish the task.

1.

2.

3.

4.

5.

6.

7.

8.

9.

STEP #3 BRAINSTORMING

List some possible solutions to the challenge.

1.

2.

3.

4.

5.

6.

7.

8.

STEP #4 ROBOT DESIGN

Use scratch paper to sketch ideas for the robot, and then choose the “best” design idea and illustrate it NEATLY below. Include any labels or explanations necessary to make your design understandable.

STEP #5 PROGRAM FLOWCHARTING

Outline the programming steps for your robot to accomplish the task. This can be in the form of a chart or graph.

STOP!!! – GET TEACHER APPROVAL BEFORE MOVING ON: ______________________

STEP #6 ROBOT BUILDING AND PROGRAMMING

Build the robot and program it according to your plan!

Design Challenge Rubric

	Extra Credit	A	B	C	F
Successful Solution	Meets criteria and one or more super challenge criteria	Solution clearly solves the problem but not super challenges.	Solution solves problem inelegantly or inefficiently.	Solution does not completely solve problem.	No reasonable attempt made to solve problem.
Programming	Program uses advanced techniques including Boolean logic, Complete palette blocks, etc. Program demonstrates extraordinary creativity or unique way of solving problem	Program is straightforward and efficient, and uses appropriate programming constructs. Program has a reasonable algorithm for solving problem and uses good logic.	Program is straightforward and easy to understand. Program is inefficient. Program has a reasonable algorithm for solving problem.	Program is poorly written or difficult to understand. Program has unused parts or does not correctly control robot. Algorithm is strained.	Program does not work. Program does not solve problem effectively.
Construction	Robot constructed using advanced gearing or other advanced construction techniques. Robot demonstrates extraordinary creativity.	Robot is of sound construction: nothing falls off, robot works as intended. Mechanics used well to achieve desired outcome. Robot can solve problem repeatedly.	Robot works as intended, but some extraneous parts fall off. Moderate degree of repeatability: robot will run again but must be adjusted or fixed.	Robot does not work as intended, but does move. Robot falls apart. Very simple construction – mechanics not used well. Robot cannot run repeatedly.	Robot falls apart or does not move at all. Construction appears careless or haphazard.
Documentation	Documentation goes beyond required paperwork.	Ample and accurate documentation. Documentation kept consistently and thoroughly.	Good documentation: documentation kept consistently but not as thorough as it could be.	Fair documentation: documentation kept inconsistently and missing parts.	Little or no documentation
Cooperation	Student(s) helped other groups	Student worked well with group. Student participated actively in all parts of project.	Student worked somewhat well with group. Student participated in most parts of project.	Student had trouble working with group. Student participated in few parts of project.	Student did not participate in project. Student sabotaged others' work.

Daily Group Evaluation Date: ___________________

List each member of your group (including yourself) and assess each area with:

3 = strongly agree (s/he was very good at this)

2 = agree (about right)

1 = disagree (this was a problem)

Name	*Listened respectfully to group members*	*Was focused and on-task*	*Did his/her share of work*
(self)			

Comments:

Information Specialist Report

You are responsible for reporting the status of the project to the Team Manager every day. How has the team progressed? Address the following questions:

1. What did your team accomplish today?
2. What problems did the team find today?
3. What solutions did the team try?
4. Other comments?

	Tasks	**Report**
Per 1	Get Challenge Begin brainstorming & Designing	
Per 2	Finish Design & get approval Begin building test parts—try different ideas	
Per 3	Finish building test parts & begin assembling robot from successfully tested parts	
Per 4	Continue assembling robot from parts Create program for robot	
Per 5	Continue building & programming robot—test regularly	
Per 6	Continue to refine robot—test regularly with the program	
Per 7	Finish refining robot—make sure it completes challenge!	
Per 8	Finish or enhance robot	
Per 9	Design Challenge: Show off robot!	
Per 10	Clean up: Take apart robot, return materials to original state	

Names: ____