University of Wisconsin-Extension

4-H Youth Development

I DO Robotics

LEGO® Robotics Educator Guide

LEGO® Mindstorms® NXT

Developed by:

Dr. Joanna M. Skluzacek
4-H STEM Specialist

AND

Heidi Dusek
Youth and Family Educator
Outagamie County

Introduction to LEGO® Mindstorms® NXT v 2.1

This curriculum packet is an introduction to LEGO® Mindstorms® NXT robotic system with the v 2.1 programming software. The curriculum can assist even the most novice instructor in gaining experience and confidence in robotics and help advanced instructors to easily implement new challenges into their club or group setting.

The goals of this guide are to:

- Provide background and research on how robotics' projects help youth develop STEM (science, technology, engineering and math) content and life skills,
- Summarize essential information about LEGO® Mindstorms® NXT 2.1 robotic system,
- Assist in troubleshooting technology problems and,
- Provide detailed instructions and solutions for robotic challenges.

<u>Why Robotics</u>

A recent study by the President's Council of Advisors on Science and Technology found: "Economic forecasts point to a need for producing, over the next decade, approximately 1 million more college graduates in STEM fields than expected under current assumptions." The shortage of skilled STEM workers has been documented in several research papers (Bayer Corp., 2012; Fadigan & Hammrich, 2004; Grasso, Callahan, & Doucett, 2004; Smith, Heck, & Worker, 2012; Tran & Nathan, 2010) and government reports (Beatty & National Research Council, 2011; Committee on Prospering in the Global Economy of the 21st Century & Committee on Science Engineering and Public Policy, 2007; National Research Council, 2011; Nielsen & National Research Council, 2011). One proposed solution to this issue is to engage youth at a young age and throughout their formative years in STEM programs and projects, while inspiring them to seek out job opportunities within a STEM field. Teaching youth STEM through robotics allows the student to learn content knowledge of a subject area by applying the content in a real-world context. There are many examples of how robotic systems are used daily and this connection between educational programs and real-world skills enables students view the relevance in their educational programs.

Studies specific to LEGO® Mindstorms® NXT suggest youth gain problem solving skills, creative thinking skills and an increase in scientific-technological knowledge (Barak & Zadok, 2009; Barker, 2007; Chen Yuan, 2012; Lindh & Holgersson, 2007; Mosley & Kline, 2006; Nagchaudhuri, 2002; Slangen, van Keulen, & Gravemeijer, 2011). Other studies show youth who worked with LEGO® Mindstorms® NXT systems improved in-school performance in STEM related disciplines (Cejka, Rogers, & Portsmore, 2006; Panadero, Romaacuten, & Kloos, 2010). The inherently interdisciplinary nature of the Mindstorms® robotic systems encourages participants to use different types of skills and has proved to capture youth in long-term engagement which increases content mastery. Since the robot is created from LEGO® building blocks, youth have an almost infinite array of configurations they can build, inspiring creativity and increasing engagement. The incorporation of computer programming into the project allows participants to become more comfortable with technology and computer operations. The software of the Mindstorms® NXT robot is based on the LabVIEW platform, which is the industry standard in computer assisted data collection and experiment design. This makes the skills gained by using the software transferable to the workforce and higher education.

Alignment with 4-H Youth Development Outcomes

This curriculum is meant to serve as a resource for adult and youth leaders in robotics. The anticipated outcomes of using this resource are:

Adults will increase their comfort and confidence in leading STEM-based activities.

Adults will gain familiarity with the experiential learning model.

Youth will increase their involvement in 4-H STEM project opportunities.

Youth will demonstrate creativity, innovation and critical thinking skills.

> Youth can justify their decisions about their programming strategy.
>
> Youth can use information gained through trial-and-error programming to refine decision making skills.
>
> Youth can develop their own robotic challenges and express creativity and critical thinking.
>
> Since there is not one right way to complete a challenge, youth can be creative in their robotic program.

Youth will increase their ability to work collaboratively with others.

> Youth will work in pairs or teams to complete challenges.
>
> Youth will respect others opinions on how to solve problems.

Youth will be able to improve communication skills.

> Youth will be able to reflect on successes and failures in programming and describe the issues.
>
> Youth will use technology to help express ideas.
>
> Youth will learn how to deal with stress and trial-and-error in programming.

Alignment with 4-H Science Abilities, 4-H Science Process Skills and STEM Content Expertise

The 4-H Science Abilities are divided into overarching process skills. These seven process skills are: observing, questioning, hypothesizing, predicting, investigating, interpreting, and communicating. Several of these skills are described in the previous section. In addition to life skill development outcomes, youth who use robotics as a platform for learning also gain abilities specific to STEM content. For instance, youth will observe how the robot performs, question why the robot performed as it did and troubleshoot the program using the information gathered. Specifically, participants will improve their basic computer operation skills (save and open files, cut and paste tools); gain experience with application of basic math functions (addition, subtraction, etc.); apply geometry to calculate turn radius and wheel rotation distance; and use units of measurement along with measurement estimation and comparison.

How to Use This Curriculum Guide

This curriculum guide is designed to be used with the LEGO® Mindstorms® NXT base set robot and the LEGO® Mindstorms® NXT software v 2.1 purchased from the LEGO® online store (http://www.lego.com). There are other NXT robots available through other vendors, but some of these kits do not have the same components as the base set LEGO® kit. The part number for the Mindstorms® NXT robot is EE979797 while the software part number is EE900080.

The LEGO® Mindstorms® NXT robotic kits are available for check-out through the University of Wisconsin-Extension Media Library for a nominal fee. Many Wisconsin regions also have robotic kits and laptops available for lending. It is recommended to have one robot and computer (loaded with the NXT software) per two youth participants.

The lessons are created to be used with the "basic robot" configuration. This means the robot is built according to the instruction manual that comes with the base set kit. The building instruction manual provides stepwise procedures for building, along with length scale keys so youth can easily find the right sized part by comparing it to the picture. Furthermore, all the lessons included in this guide are created using the common palette screen of the software. See Figure 3 for further information regarding programming palettes.

The curriculum provided in this guide is for the instructor; however, the guide may be edited for youth participants depending on the teaching strategy, objectives and venue of the program. For example, the instructor can give the participants the challenge instructions as text only, an abbreviated programming guide or a full programming guide. If participants are given the text only, the instructor will need to have more direct teaching time to describe the programming blocks and the details of downloading the program from the computer to the robot. Each lesson is given a difficulty rating in stars. The scale is from one star to five stars, with one star indicating a lesson that is easy and five stars indicating difficult.

The full pictorial programming guide can be given to youth directly, however this in only recommended if the instructor has very little time for the session, for instance, if the program is a marketing event for a robotics camp or club project. It might also be helpful to give youth the full programming guide for the first activity as part of a longer session. In this instance, the instructor should take time for the group to discuss each programming block and the variables associated with the programming, since the youth will not develop this understanding by using the complete pictorial guide alone. In most cases, the instructor will use the text only or a subset of the abbreviated programming guide.

The abbreviated and complete pictorial guides provide examples of solutions to complete the challenges. It is important to remember that there are multiple ways to successfully complete challenges and youth are encouraged to be creative with their programming. It is also important to recognize that the numbers for duration, etc. provided within the complete pictorial guide are estimates. The robot will perform differently under different conditions. Examples of performance variables are described in the Distance and Turns Instructor Guide.

Essential Information about Mindstorms® NXT Robots

Before the youth begin building the robot, **check page two** of the building instruction manual to view part replacements. One part routinely causes problems in the building process; part 655826. Throughout the building guide the part is black, however; it has been replaced with a blue part in the kit.

In the basic robot configuration, it is important for the **rear wheel** (as described on pages 16-17 of the building instruction manual) to freely rotate. If the wheel does not rotate all the way around freely it might be on backwards or otherwise incorrectly built. Check the building instructions to correct the issue.

Make sure the **battery** is charged. The rechargeable battery is reliable and does last a long time (at least 15 hours) when charged, however; if the robot has been left unused for a long period of time or exposed to cold weather during transport or shipping the battery might be discharged.

Having **AA batteries** on hand may be helpful if the battery is not working properly.

When using the **ultrasonic sensor** remind the youth to not have their hands in front of the sensor when they press the orange button to start the program. The sensor will sense their hand and the robot will immediately travel backwards.

The **ultrasonic sensor** has also been known to act erratically because the sensor becomes corroded after two to three years. Replacing this sensor after two years of use is the only way to avoid this problem.

There are two ways to build the **light sensor** in the building manual. In one configuration the light sensor is pointing straight forward and in the other configuration the light sensor is pointing down. Make sure to select the correct configuration for the task.

Trouble Shooting Tips

The following are some errors that participants occasionally encounter when using the Mindstorms® NXT v 2.1 software. Most of the errors are encountered when trying to download the program from the computer to the robot and will look something like this:

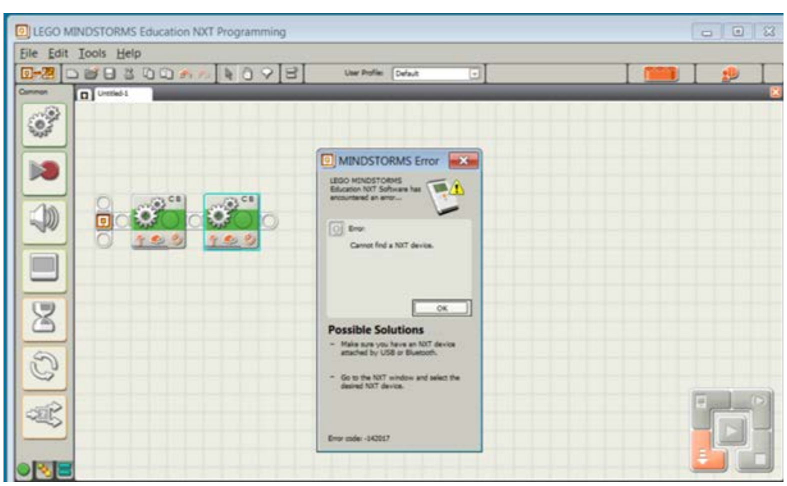


Figure 1: Cannot find NXT device error screen.

The most common error screen that will come up is the “**Cannot find a NXT device**”. This means the robot is not connected to the computer through the USB port (or Bluetooth) or that the NXT is not turned on. Check the connections to make sure they are secure or try a different USB port on your computer.

Another error screen one might see is the “**NXT is out of memory**”. Before proceeding files will have to be deleted off the Mindstorms® robot to make room for the new program. This can be accomplished by removing old programs or deleting sound files that are rarely used. There is no need to worry about deleting sound files off the NXT robot since one can easily restore the NXT to factory settings and presets.

One can also check the NXT memory functionality by connecting the NXT robot to the computer and selecting the NXT window icon as shown in Figure 2.

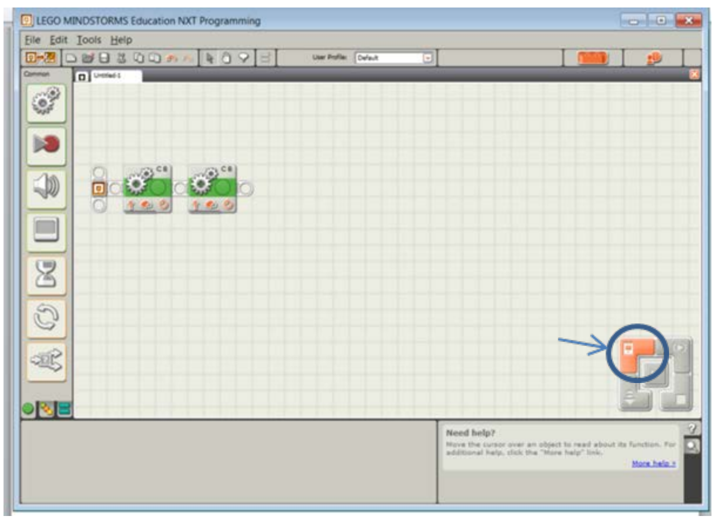


Figure 2: Area to click to show the NXT window icon.

Selecting the NXT window will give you lots of useful information about your NXT.

If the software version on the computer is not the same as what is installed on the NXT, an error message stating “**Firmware out-of-date**” will pop up on the computer screen. The error screen gives instructions for resolving this problem. Connect the robot to the computer via the USB cord (you cannot install firmware using the Bluetooth connection). From the top menu, select Tools →Update NXT Firmware. Once selected a download screen will come into view. Select “download”. An internet connection is not needed to download the firmware.

Unlike the easily fixed errors listed above, occasionally one might see “**The required file is broken**” error screen. Unfortunately, the only sure way to remedy this error is to close and restart the software program. The program can be saved and used for reference, but the participants will have to recreate the program.

Occasionally, a participant might accidentally switch from the **common palette to the complete palette**. To return to the common palette, click on the tab buttons under the programming blocks on the left side of the screen as displayed in Figure 3.

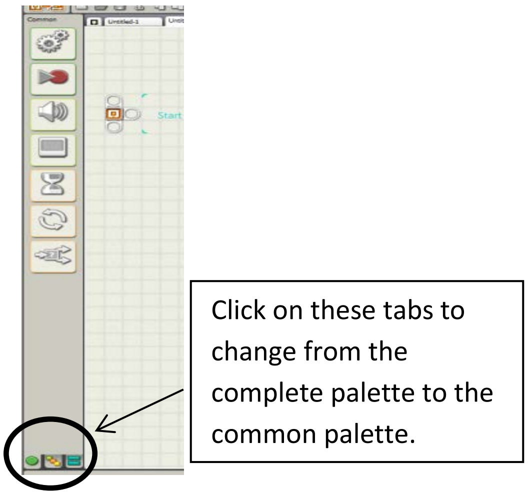


Figure 3: Location of tabs to select for the Common, Complete or Custom Palettes.

Like all electronics, sometimes the software might malfunction or the NXT might begin to perform erratically. These random errors can usually be fixed by shutting down and restarting the computer and/or turning the NXT off and then on.

If it seems that the NXT is continually having problems, one can completely **reset the NXT brick to factory settings** by accessing the reset button with a paper clip as shown in the following picture.

Figure 4: Location of the reset button to return NXT brick to factory settings.

Using the paper clip, hold down the reset button for five to seven seconds while the NXT is in the on position. The NXT will shut down and when restarted the NXT will make a clicking sound. Connect the cleared NXT brick to the computer and install the firmware as described above. This procedure can also be used as a preventive measure to prepare for an event.

<u>Resources</u>

This curriculum guide is meant to serve as an introduction to robotics and provide a few robotic challenges to get instructors started using this engaging learning environment. However, LEGO® Mindstorms® robotic systems are widely used and there are several other places to find teaching resources. One important resource is a tutorial entitled “**Robot Educator**” that is embedded into the NXT software program. You can access this tutorial by clicking on the orange LEGO piece in the upper right hand corner of the software program as seen in Figure 5.

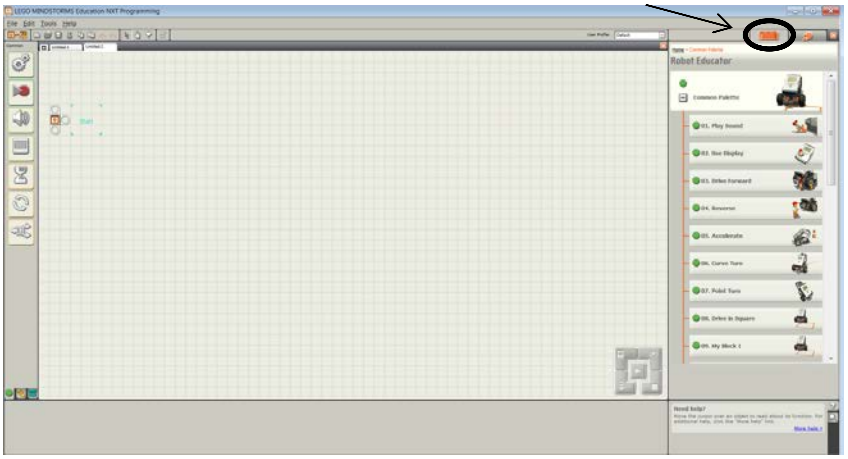


Figure 5: Area to click to select the Robot Educator Tutorial Guide.

There is also a video tutorial developed by Dr. Joanna M. Skluzacek which discusses how to program each sensor of the NXT. The video segments are available on the following website: http://fyi.uwex.edu/4htechwizards/mentors/resources/video-resources/ or in a DVD format from the Wisconsin State 4-H office.

Below is a list of peer-reviewed teaching resources:

http://www.legoeducation.us/eng/competitions/

http://www.legoeducation.us/eng/characteristics/ProductLine~LEGO%20MINDSTORMS%20Education%20NXT

http://www.wichita.edu/thisis/home/?u=mindstorms

http://firstlegoleague.org/challenge/thechallenge

http://4hset.unl.edu/4hdrupal/

http://www.nxtprograms.com/index2.html

References

Barak, M., & Zadok, Y. (2009). Robotics projects and learning concepts in science, technology and problem solving. *International Journal of Technology and Design Education, 19*(3), 289-307.

Barker, B., & Asnorge, J. (2007). Robotics as Means to Increase Achievement Scores in an Informal Learning Environment. *Journal of Research on Technology in Education, 39*(3), 229-243.

Bayer Corp. (2012). Bayer Facts of Science Education XV: A View from the Gatekeepers-STEM Department Chairs at America's Top 200 Research Universities on Female and Underrepresented Minority Undergraduate STEM Students. *Journal of Science Education and Technology, 21*(3), 317-324.

Beatty, A. S., National Research Council. Committee on Highly Successful Schools or Programs for K-12 STEM Education, National Research Council. Board on Science Education & National Research Council. Center for Education. Board on Testing and Assessment. (2011). *Successful STEM education : a workshop summary*. Washington, D.C.: National Academies Press.

Cejka, E., Rogers, C., & Portsmore, M. (2006). Kindergarten robotics: Using robotics to motivate math, science, and engineering literacy in elementary school. *International Journal of Engineering Education, 22*(4), 711-722.

Chen Yuan, C. (2012). An innovative knowledge management learning cycle by Lego NXT for science education. *International Journal of Innovative Computing, Information & Control, 8*(1B), 791-798.

Committee on Prospering in the Global Economy of the 21st Century & Committee on Science Engineering and Public Policy. (2007). *Rising above the gathering storm : energizing and employing America for a brighter economic future*. Washington, D.C.: National Academies Press.

Fadigan, K. A., & Hammrich, P. L. (2004). A longitudinal study of the educational and career trajectories of female participants of an urban informal science education program. *Journal of Research in Science Teaching, 41*(8), 835-860.

Grasso, D., Callahan, K. M., & Doucett, S. (2004). Defining engineering thought. *International Journal of Engineering Education, 20*(3), 412-415.

Lindh, J., & Holgersson, T. (2007). Does lego training stimulate pupils' ability to solve logical problems? *Computers & Education, 49*(4), 1097-1111.

Mosley, P., & Kline, R. (2006). Engaging students: a framework using LEGO robotics to teach problem solving. *Information Technology, Learning, and Performance Journal, 24*(1), 39-4545.

Nagchaudhuri, A., Singh, G., Kaur, M., George, S. (2002). *LEGO Robotics Products Boost Student Creativity in Pre-College Programs at UMES*. Paper presented at the ASEE/IEEE Frontiers in Education Conference, Boston, MA.

National Research Council. Committee on Highly Successful Schools or Programs for K-12 STEM Education, National Research Council. Board on Science Education, & National Research Council. Center for Education. Board on Testing and Assessment. (2011). *Successful K-12 STEM education: identifying effective approaches in science, technology, engineering, and mathematics*. Washington, D.C.: National Academies Press.

Nielsen, N. & National Research Council. Planning Committee on Evidence on Selected Innovations in Undergraduate STEM Education. (2011). *Promising practices in undergraduate science, technology, engineering, and mathematics education : summary of two workshops*. Washington, D.C.: National Academies Press.

Panadero, C. F., Romaacuten, J. V., & Kloos, C. D. (2010). Impact of learning experiences using LEGO Mindstorms reg in engineering courses *2010 IEEE Education Engineering 2010 - The Future of Global Learning Engineering Education* (pp. 503-512)

Slangen, L., van Keulen, H., & Gravemeijer, K. (2011). What pupils can learn from working with robotic direct manipulation environments. *International Journal of Technology and Design Education, 21*(4), 449-469.

Smith, M. H., Heck, K. E., & Worker, S. M. (2012). 4-H boosts youth scientific literacy with ANR water education curriculum. *California Agriculture, 66*(4), 158-163.

Tran, N. A., & Nathan, M. J. (2010). The Effects of Pre-Engineering Studies on Mathematics and Science Achievement for High School Students. *International Journal of Engineering Education, 26*(5), 1049-1060.

Teambuilding: The Robot Drawing Game

REQUIREMENTS
Number of Players:
Any (must work as partner team)

Ages: 8 and up

Activity Level: Medium

Equipment:
Computer Paper
Pencils
Poster board or dry erase board (not in learning area)

NOTES:
Level of Difficulty can depend on the age of the students. Final product can be entire diagram/drawing, or step by step process of making the diagram/drawing If you have an odd number try to assign an adult as partner or give them the responsibility of creating the diagram.

Reference:
The Robot Drawing game was modified by Heidi Dusek. Original Source was *Partner Drawing Game,* Project Lead the Way: 2012 Gateway Academy Curricula (www.pltw.org).

DESCRIPTION:
As a partner team your goal is to produce a replica of the drawing in the hallway (or where ever the finished drawing will be). Each team member has a very specific role. As a team you must work together to give directions and listen to create the final product. The final product does not depend on their skills as "runners" or "drawers," but on their ability to communicate effectively.

Procedure:

1. Group students into partner groups (2 people per team)
2. Give each team 1 piece of paper and 1 pencil.
3. Determine who is going to be the "runner" and who is going to be the "drawer."
4. The Drawer should have the pencil and paper.
5. Partner teams should sit back to back. At no time should the runner be able to see the drawer's paper.
6. Explain that the runner will go look at the diagram/drawing and need to come back and verbally explain how to draw the diagram on the partner's paper. The runner needs to remain back to back with their partner while they are giving directions. The runner cannot touch the pencil, paper or see that drawing until the activity is over.
7. The drawer must drawer what they are told by the runner, no matter how confusing. The drawer does not have the ability to ask questions.
8. Give the team 2 minutes to complete the drawing and then show their drawings.
9. Have the partners switch roles and repeat steps 1-6 with new roles.
10. Once each partner has played both roles take some time to process the activity as a large group.
    Discussion Questions:
    What was hard about this activity?
    What would have made it easier?
11. *Those who were runners the first time, How do you feel the game was going? Did you think you were giving good directions?*
12. *Those who were the runners the 2^{nd} time, what did you try to do differently to be successful as a team?*
13. Sometimes what makes sense to us, doesn't make sense to others. If we think of our partners as robots who don't know what "Big" or "Small" is. How could we give directions that would be clearer?
    Give them a reference: What way should the paper be? What is the size relative to: a quarter, an orange, a basketball? How could

we measure distance: knuckle, finger, hand, etc.

14. Repeat the activity one more time, however this time allow the person drawing to ask questions to clarify.
15. Once complete review as a large group
16. *Discussion Questions:*
17. *Which way was easier?*
18. *How did it feel to have to follow directions without being able to ask questions?*

Conclusion and Reflection

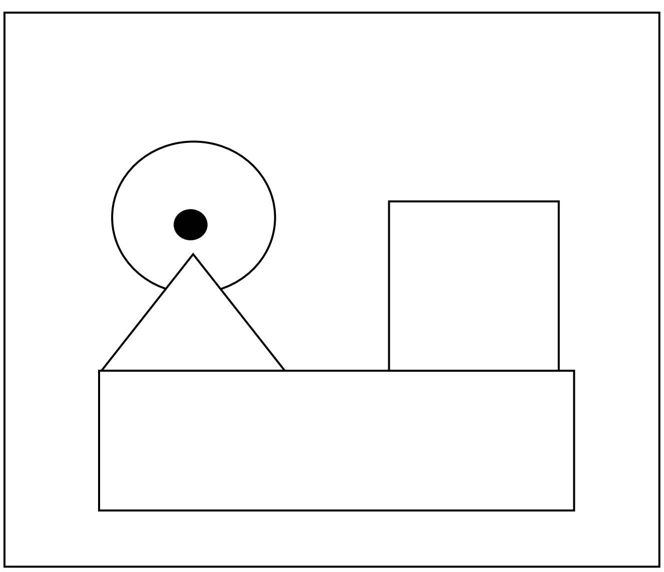
How does this activity relate to real life?

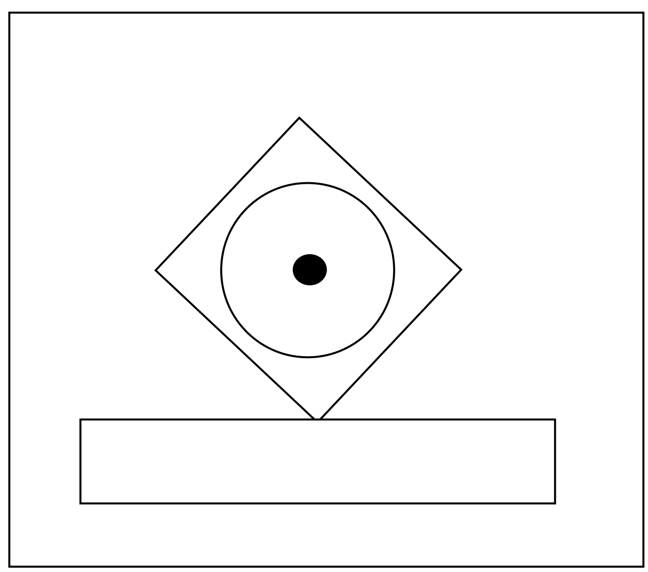
(*one-way communication is similar to being lectured or yelled at, being told what to do and not being listened to; two-way communication makes talking and listening equally important. Good communication is taking turns to talk and listen to make sure you understand others perspectives)*

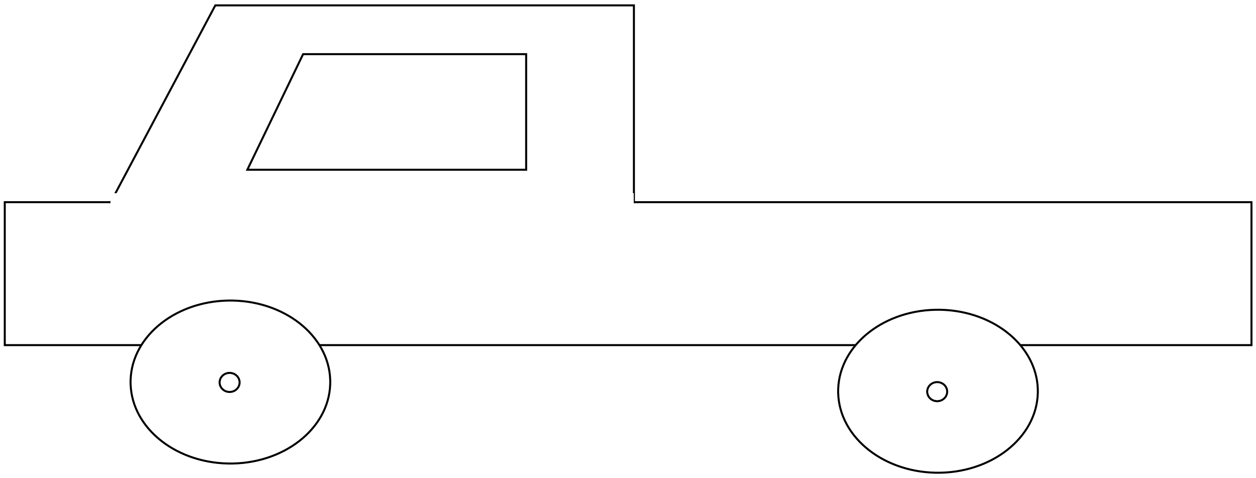
How does this activity relate to building and programming robots?

(Robots don't have brains and therefore can't think for themselves. A robot will only do what we tell it to do, so the communication is one-way. If we want a robot to do a specific task, we have to give it very clear directions.

Sample Diagrams/Drawings







Teambuilding: Group Spell

REQUIREMENTS	DESCRIPTION
REQUIREMENTS Number of Players: Any (must work in small groups) Ages: 8 and up Activity Level: Medium Equipment: None Reference: The Group Spell Activity was modified by Heidi Dusek. Original Source was *Teambuilding Activities: Southern District Leadership Team, pg. 9, Team Body Spelling (2002)* http://4h.uwex.edu/clubs/documents/teambuilding0231.pdf	DESCRIPTION: In this activity the leader will give direction as to what part of the body to use to spell out a word. Youth are divided into small groups of 4-8 people. All groups must achieve success by spelling out the word, and group leader verifies. Procedure: 1. Divide youth into groups of 4-8 youth per group. 2. Explain the goal of the game is for every group to achieve success. Success means that everyone in your group participated by using the assigned body part to spell the word. 3. Once groups have formed and youth know the other members of their group shout out the assigned body part. Some examples might be: index finger, right arm, both legs, entire body, left hand, etc. 4. Then shout out the word that you'd like them to spell. Try to keep the words related to the content you are learning, in this case vocabulary words that relate to robotics: Robot, teamwork, program, build, gears, sensors, etc. 5. Walk around to identify strategies groups are using to be successful. How are they communicating, do all the groups look the same? 6. See Notes for additional ways to adapt this activity. Notes: 1. Start with smaller sized groups, then combining groups to

	make larger groups as the game goes on. 2. Mix up the groups each time. Call out how many should be in a group at the beginning of each round. Encourage kids to work with different groups of people. 3. Have the groups decide on a word that describes something they all have in common and spell it out using the assigned body part. 4. Have the groups identify how the word relates to the content they are learning (definition). 5. Before assigning them the word to have each group spell, give the instructions to work in silence to create the word. **Conclusion and Reflection** Did every group spell the words exactly the same? Why or why not? *Usually not, because we all draw on our strengths and experiences to find a solution that makes sense to us. We can all achieve the solution using different methods.* Would you have been successful doing this activity by yourself? Why or Why Not? *We were just one piece to the puzzle. We often need the skills and resources of others to complete a task, which is true in many industries as well. Even some of the most successful people in the world work with other members of their team to find solutions.*

Instructor Guide
Title: Distance and Turns

Introduction
Calculating the distance the robot will travel for each of the duration variables (rotations, degrees, seconds) can be confusing for participants especially when coupled with a turn or a spin. It is important to remember that rotations and degrees reference the wheel axle, such that these distances can change depending on the size of the tire installed on the robot. This activity outlines how to use the duration variables correctly to make corner turns and to calculate the distance to objects using the circumference of the tire. The answers provided in the instructor guide are based on the basic bot construction from the LEGO® Mindstorms® v 2.0 kit (part number EE979797).

Objectives
Youth will apply basic math functions and geometry.
Youth will gain experience with units of measurement and measurement comparison.
Youth will demonstrate creativity, innovation and critical thinking skills.
Youth will increase their ability to work collaboratively with others.
Youth will better understand the process of programming and evaluating robotic movements.

Preparation and Materials
One-12 inch ruler per robot

Activity: How far will your robot travel?
Take a look at the larger tires connected to the gear motors on the NXT robot. On the side of these tires there are numbers. This is true for *all* tires and tells you the size (diameter and width) of the tire. Automobile tires and bicycle tires also have these numbers to ensure the mechanic places the correct tire size on each axle.

1. What are the numbers on the robots large tire?

   56 X 26

2. Are these numbers are in inches, centimeters or millimeters?

   Millimeters

3. Which number is the diameter and which is the width?

   56 mm = diameter

   26 mm = width

4. Using the diameter of the tire, one can find the *Circumference* of the tire. The circumference of a tire tells the distance a tire travels in one revolution:

   $$C = \pi D$$

   Where C= circumference, π = 3.14, and D = diameter

   $$\begin{array}{r} 3.14 \\ \underline{\times\ 56} \\ 1884 \\ \underline{15700} \\ 175.84 \text{ mm} \end{array}$$

   What is the circumference of the NXT tire?

This means one rotation of the tire is 175.84 mm or 17.6 cm.

5. If one rotation is equal to 17.6 cm, how many cm will the robot travel in three tire rotations?
   17.6 X 3 = 52.8 cm

6. Since the wheel of the robot is a circle, one can also speak about degrees of tire rotation. How many degrees does the tire rotate for each tire rotation?
   1 tire rotation – 360 degrees

7. How many cm will the robot travel if it is programmed to travel a duration of 720 degrees?
   720 degrees = 2 tire rotations = 35.2 cm

However, the robot might not travel the distance you calculated. There are several variables that affect the distance travelled by the robot:

- The **power level setting** will affect distance traveled when using the time interval in seconds; however, it will not affect the distance travelled in rotations or degrees. In addition the battery life levels will also impact the amount of power provided to the robot to complete the moves.
- The external environment will impact the amount of **friction** on the tires. A smooth surface will have less friction meaning the robot will travel slightly faster. A carpeted surface will have more friction meaning the robot will travel slightly slower.
- The **mechanics** of the built robot impacts the function of the robot. For example if a tire is not perfectly aligned on the wheel well or on the axle you may experience a wobbly wheel.
- The **temperature** will affect the distance travel. Since the wheels are made out of rubber, they will slightly expand when the temperature is hot and contract with the temperature gets colder.

Activity: Making Turns and Spins

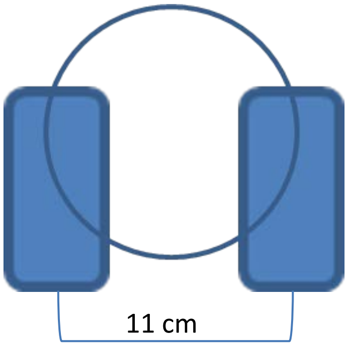
The robot will turn corners by altering the steering and duration that the wheels turn. If one wants to turn a 90 degree corner, we know from the discussion above that it is not correct to set the duration the robot travels to 90 degrees. This setting only moves the wheel a ¼ turn.

In order to understand how to make the robot turn a corner, one again needs to use some math. The wheels trace out a circle, such that if the robot makes a sharp turn, one wheel does not move and one wheel goes around in a circle of a certain diameter.

<u>Give it a try</u>
Imagine you are facing north and you want to change your direction such that you are facing east (or 90 degrees to your right). The simplest way to make this turn is to pivot your right foot and swing the left foot in a 1/4 circle.

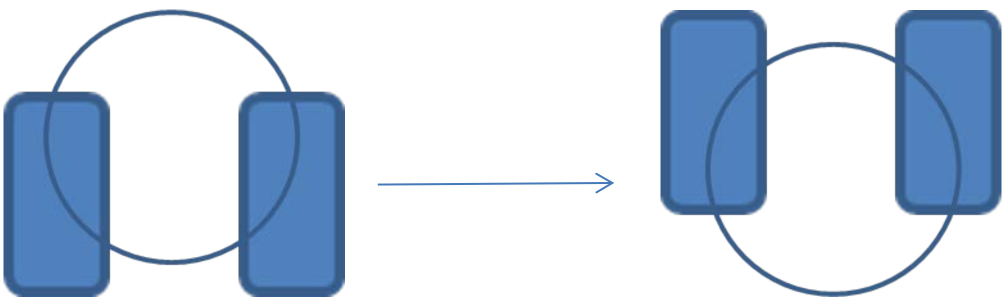
Give it a try

Stand with your feet about 10 inches apart. Pivot your right foot, but keep it on the same spot on the floor. Drag your left foot comfortably around so that you are now facing directly to the right. The distance your left foot had to travel to get to the final position is dictated by the distance between your two feet when you start. What if you started with your feet 15 inches apart? Would your left foot travel a farther or shorter distance?



The distance between our tires for the basic bot is 11 cm

Therefore, for the entire robot to spin 180 degrees:



The distance traveled by the tire = $\pi \times 11\ cm \div 2 = 3.14 \times 11 \div 2 = 17.27\ cm$

From the discussion above, one sees that 17.27 cm is close to one rotation, but not quite. Remember one rotation of the wheel will make our robot travel 17.6 cm. Thus, in order for the robot to spin around to be facing the opposite direction one can program the robot to make a sharp turn for 0.97 rotations or about 353 degrees. Of course, the wheels on the robot can move several millimeters on the axel, so this is just an estimate.

Give it a try

Program the robot to make a ½ turn (180 degree turn) and see how close to an exact turn you can come!

Instructor Guide
Challenge Title: Four Bricks

Introduction
There is no challenge mat for this activity and no sensors need to be connected to the robot. This challenge introduces participants to the move block and the relationship between the different duration types: rotations, degrees and seconds.

Objectives
Youth will demonstrate creativity, innovation and critical thinking skills.
Youth will increase their ability to work collaboratively with others.
Youth will be able to improve communication skills.
Youth will better understand the process of programming and evaluating robotic movements.
Youth will increase their ability to design a solution to a challenge.

Level of Difficulty ☆

Preparation and Materials
Instructor should determine teaching methodology (as identified in the introduction of this guide) that best fits the audience.
Print appropriate student materials.

Time Required
Programming: 5-10 minutes (if participants are given the text instructions only).
To run the course: less than a minute.

Procedure of Programming Steps
Depending on the expertise level of the participants, the instructor can give the youth the text of the steps involved in the challenge, give them the abbreviated programming guide, or give them the entire pictorial programming guide.

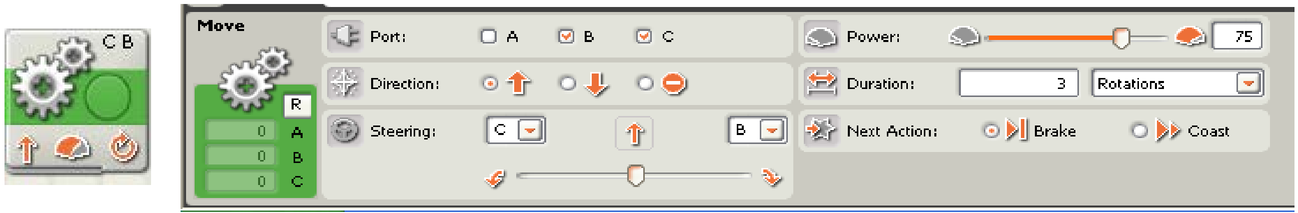
TEXT:

1. Robot moves forward three tire rotations at 75% power.
2. Robot moves backward 1080 degrees at 75% power.
3. Robot moves forward for two seconds at 75% power.
4. Robot moves forward while making a hard turn toward the right for 230 degrees at 75% power.

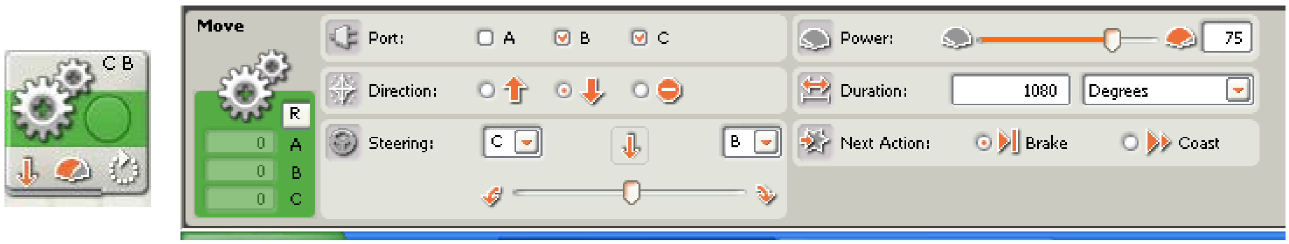
ABBREVIATED PROGRAMMING GUIDE:

COMPLETE PICTORIAL PROGRAMMING GUIDE:

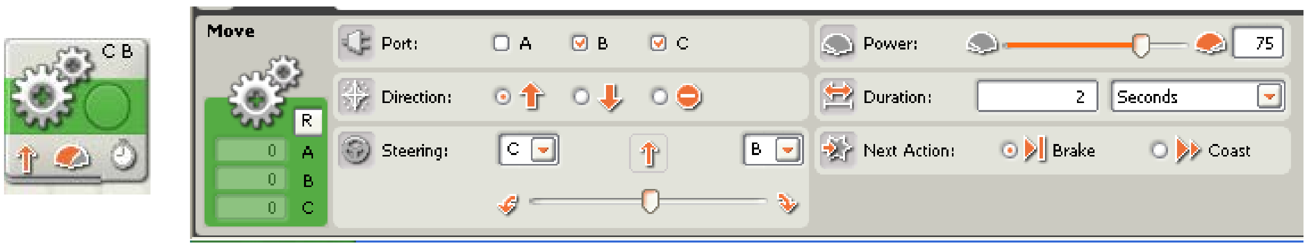
1. Robot moves forward three tire rotations at 75% power.



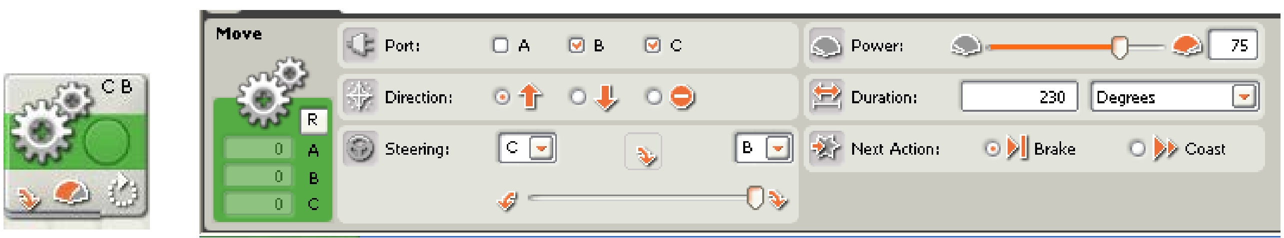
2. Robot moves backward 1080 degrees at 75% power.



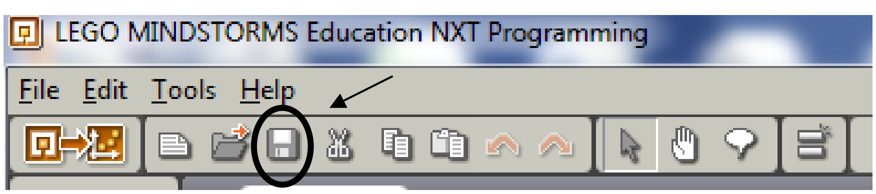
3. Robot moves forward for two seconds at 75% power.



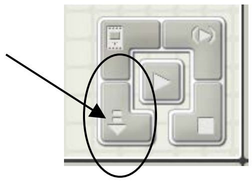
4. Robot moves forward while making a hard turn toward the right for 230 degrees at 75% power.



5. Save the computer program.



6. Download the program onto your robot using the USB cord (robot must be turned on).



7. Disconnect and use the orange button on the robot to run the program.

Discussion Questions

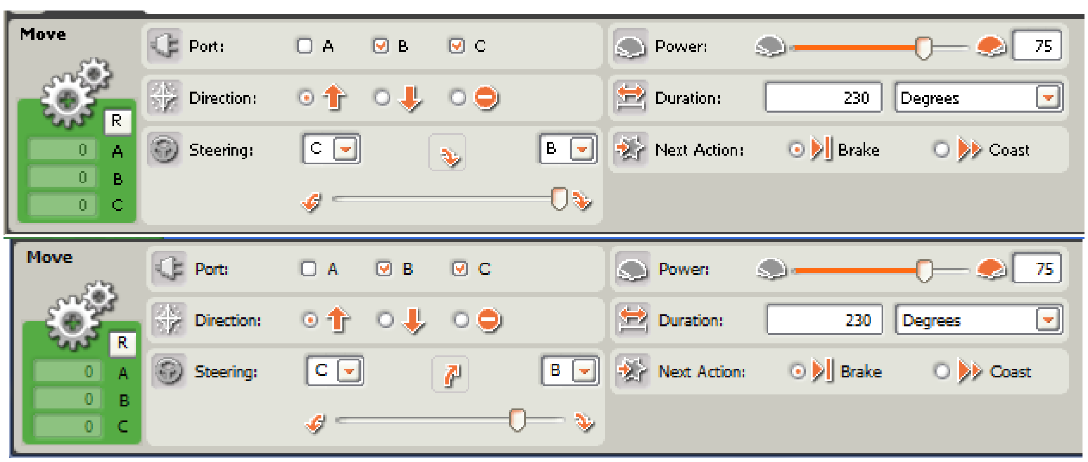
What difficulties did you encounter with the programming?

What did you do to overcome these difficulties?

How did the distance traveled by the robot in step 1 compare to the distance the robot traveled in step 2?

The distance traveled by the robot in steps 1 and 2 is the same.

What is the difference between the robot movements in the following programming blocks?



The top move program has the robot making a hard turn. This means the robot spins in place. The second program has the robot making a gentle turn. The robot will continue moving forward while turning to the right.

Instructor Guide
Challenge Title: Have a Nice Day

Introduction
This robotic challenge does not need a challenge mat. Youth can use their hand to activate the touch sensor. The robot configuration must include the sound sensor and the touch sensor to complete the challenge.

Objectives
Youth will demonstrate creativity, innovation and critical thinking skills.
Youth will increase their ability to work collaboratively with others.
Youth will be able to improve communication skills.
Youth will better understand the process of programming and evaluating robotic movements.
Youth will increase their ability to design a solution to a challenge.

Level of Difficulty ☆

Preparation and Materials
Instructor should determine teaching methodology (as identified in the introduction of this guide) that best fits the audience.
Print appropriate student materials.

Troubleshooting & Teaching Tips
If there is a lot of background noise in the room, youth may need to adjust the sound sensor level from the default of 50 to a higher number.

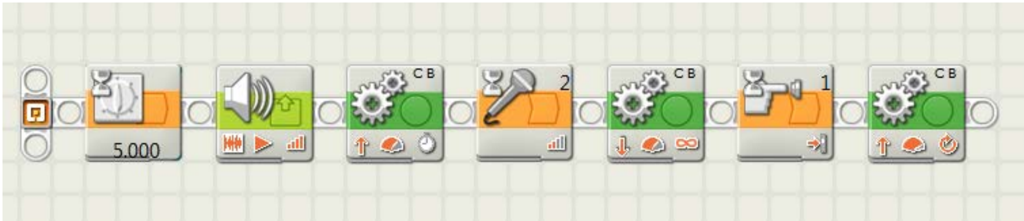
Time Required
Programming: 15 minutes (if participants are given text instructions only).
To run the course: less than a minute.

Procedure of Programming Steps
Depending on the expertise level of the participants, the instructor can give the youth the text of the steps involved in the challenge, give them the abbreviated programming guide, or give them the entire pictorial programming guide.

TEXT:

1. The robot waits five seconds before starting the program.
2. The robot says "have a nice day".
3. The robot moves forward for 3 seconds at 75% power.
4. The robot stops and waits for sound (the sound can be a clap).
5. The robot moves backwards and uses the touch sensor to hit an obstacle (youth can use their hand to activate the touch sensor).
6. The robot moves forward 3 tire rotations at 100% power.

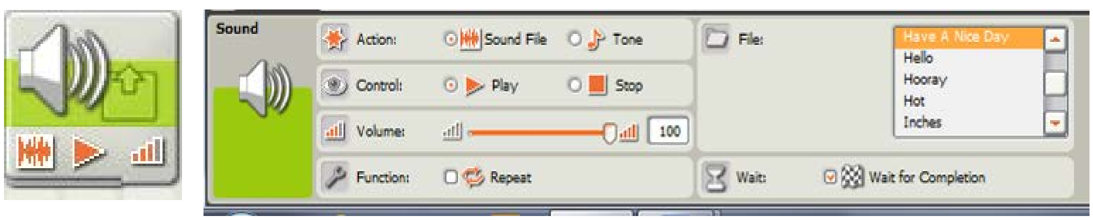
ABBREVIATED PROGRAMMING GUIDE:



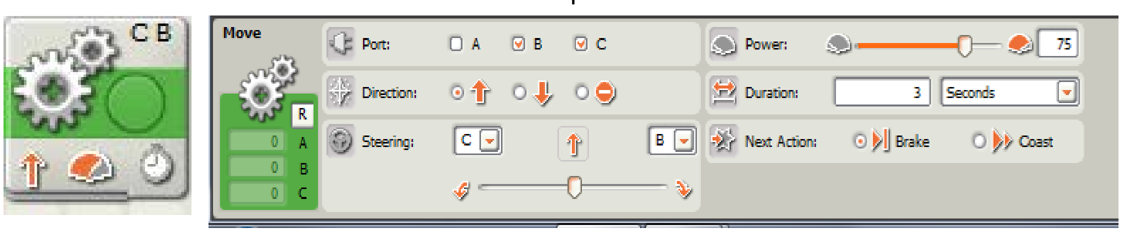
COMPLETE PICTORIAL PROGRAMMING GUIDE:

1. The robot waits five seconds before starting the program.

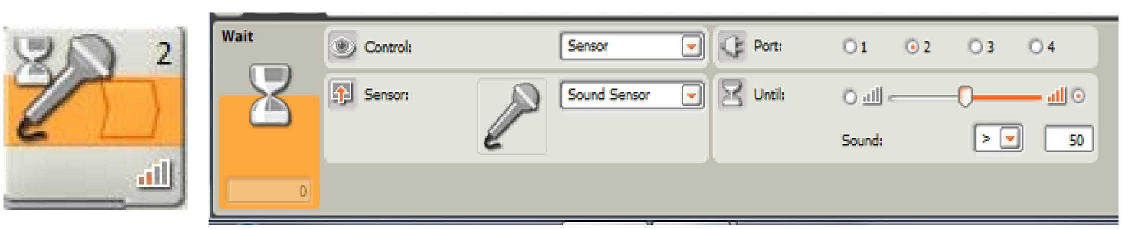
2. The robot says “have a nice day”.



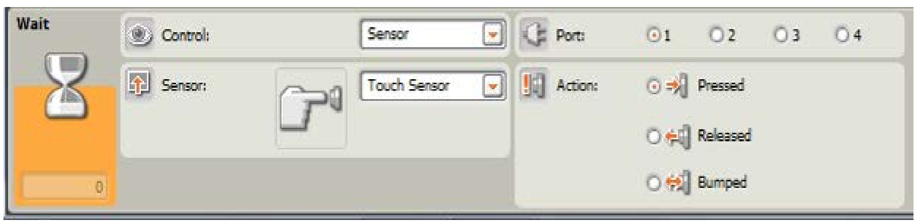
3. The robot moves forward for 3 seconds at 75% power.



4. The robot stops and waits for a sound (the sound can be a clap).

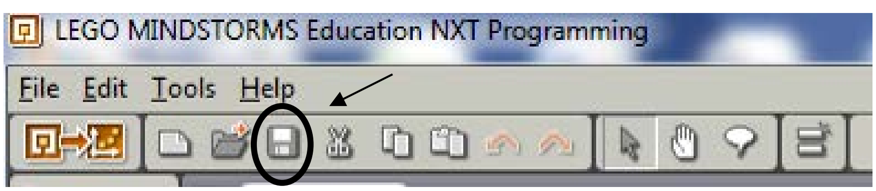


5. The robot moves backwards and uses the touch sensor to hit an obstacle (youth can use their hand to activate the touch sensor).

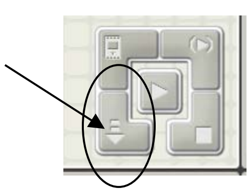


6. The robot moves forward three tire rotations at 100% power.

7. Save the computer program.



8. Download the program onto the robot using the USB cord (robot must be turned on).



9. Disconnect and use the orange button on the robot to run the program.

Discussion Questions

What difficulties did you encounter with the programming?

What did you do to overcome these difficulties?

What other things might use a touch sensor?

Automatic car wash

Touch screens

What other things might use a sound sensor?

Any voice activation software: Siri, etc.

Many new car radios will decrease sound when engine noise is reduced.

Why did you need to select "unlimited" for the move block in Step 5?

You needed to select "unlimited" because you didn't know how far away the obstacle was that would activate the touch sensor.

Instructor Guide
Challenge Title: X Marks the Spot

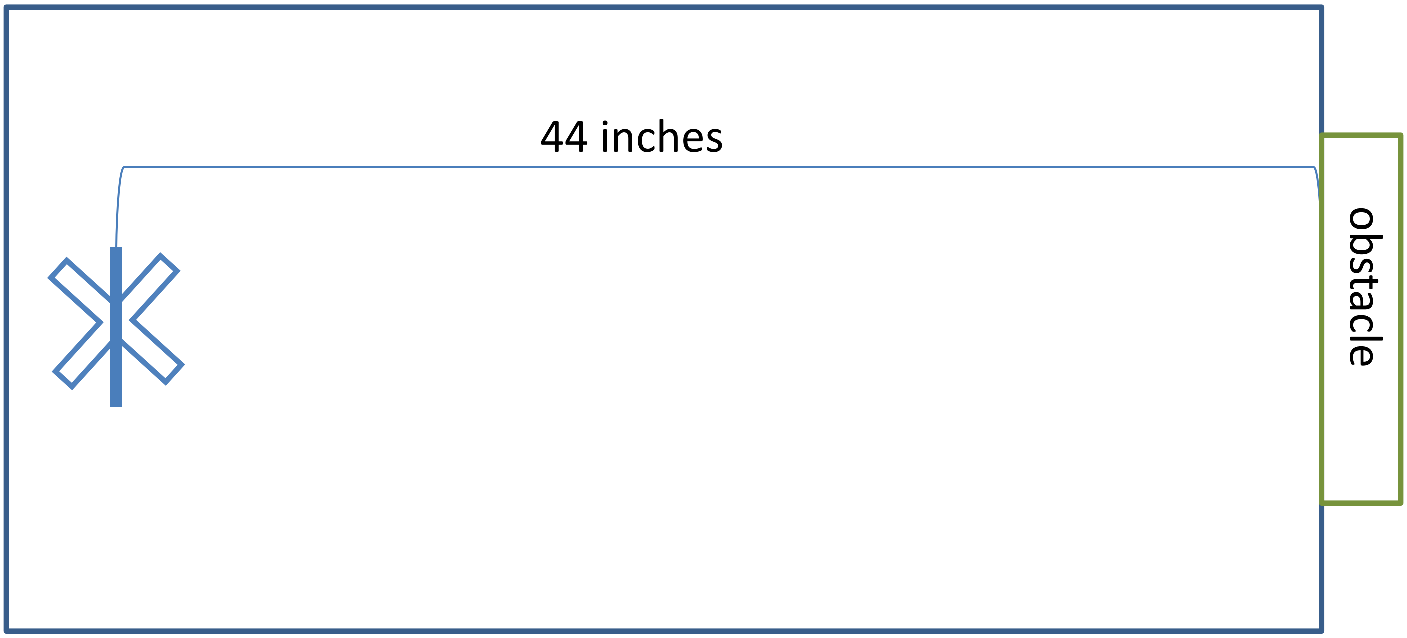
Introduction

This robotic challenge uses a challenge mat. The robot configuration must include the touch sensor and the ultrasonic (distance) sensor. The obstacle listed on the challenge schematic can be a wall or any other object that will not move when struck by the robot, but it must be at least nine (9) inches in height to activate the ultrasonic sensor (on the basic robot). The participants will not be told the distance from the obstacle to the X for steps 4-5. They will need to use trial-and-error to calculate the proper travel duration of the robot. For example, they might have to view how far the robot travels in one tire rotation and then estimate how many rotations it will take to cover the required distance. The youth will also have to use trial-and-error to figure out how to make the robot turn the correct arc distance so the touch sensor is facing the obstacle after the robot spins on the X in step 6. They may also reference "Distance and Turns" to estimate how many rotations or degrees it will take to successfully complete the challenge.

Objectives
Youth will demonstrate creativity, innovation and critical thinking skills.
Youth will increase their ability to work collaboratively with others.
Youth will be able to improve communication skills.
Youth will better understand the process of programming and evaluating robotic movements.
Youth will increase their ability to design a solution to a challenge.

Level of Difficulty ☆☆☆

Preparation and Materials
Instructor should determine the teaching methodology (as identified in the introduction of this guide) that best fits the audience.
Print appropriate student materials.
Create "X Marks the Spot" challenge mat using:
- Masking or painter's tape
- Measuring tape
- Paper or poster board (at least 50 X 12 inches in size)
- Obstacle such as a weighted box, heavy book, or wall at least nine inches in height


44 inches
obstacle


44 inches

Time Required
Programming: 45 minutes – 1 hour (if participants are given text instructions only).
To run the course: 2-3 minutes.

Procedure of Programming Steps
Depending on the expertise level of the participants, the instructor can give the youth the text of the steps involved in the challenge, give them the abbreviated programming guide or give them the entire pictorial programming guide.

TEXT:

1. The robot begins the challenge centered on the X facing toward the obstacle (ultrasonic sensor facing obstacle).
2. The robot moves toward the obstacle.
3. The robot uses the ultrasonic sensor to detect the obstacle at a distance of 12 inches.
4. Upon sensing the obstacle the robot moves backwards.
5. The robot stops in its original position (centered on the X as in step 1).
6. The robot spins in place 2 and ½ times such that the touch sensor is now facing the obstacle.
7. The robot moves toward the obstacle.
8. The robot touches the obstacle with the touch sensor.
9. The robot travels back to the X and stops when centered on the X.

ABBREVIATED PROGRAMMING GUIDE:

COMPLETE PICTORIAL PROGRAMMING GUIDE:

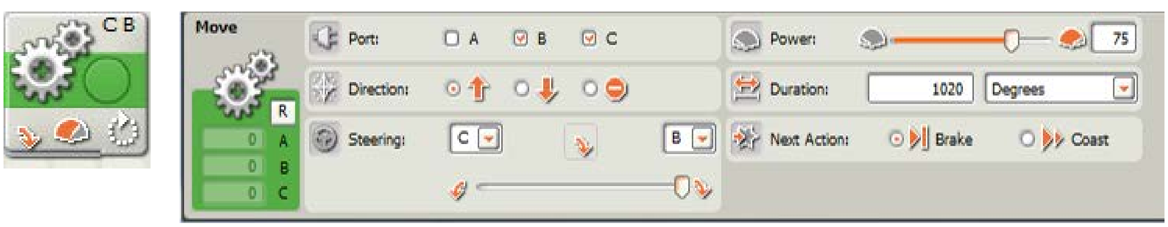
1. The robot begins the challenge centered on the X facing toward the obstacle (ultrasonic sensor facing obstacle).

2. The robot moves toward the obstacle.

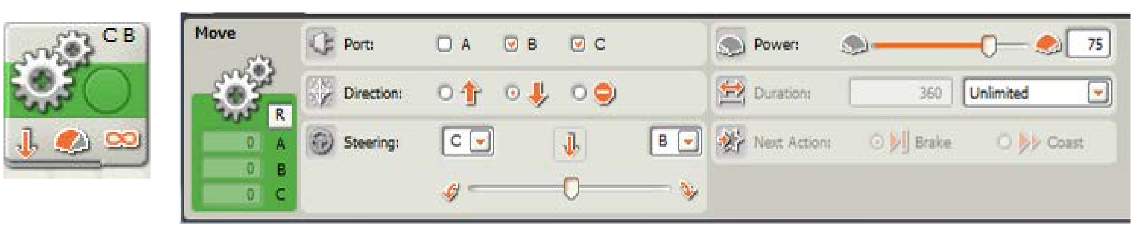
3. The robot uses the ultrasonic sensor to detect the obstacle at a distance of 12 inches.

4. Upon sensing the obstacle the robot moves backwards.
5. The robot stops in its original position (centered on the X as in step 1).

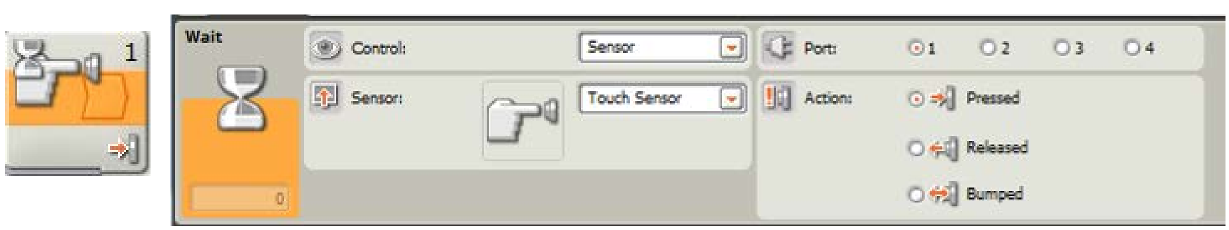
6. The robot spins in place 2 and ½ times such that the touch sensor is now facing the obstacle.



7. The robot moves toward the obstacle.

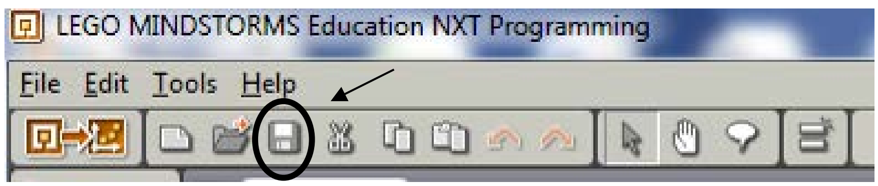


8. The robot touches the obstacle with the touch sensor.

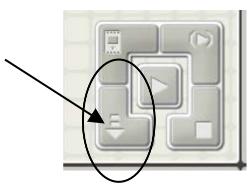


9. The robot travels back to the X and stops when centered on the X.

10. Save the computer program.



11. Download the program onto your robot using the USB cord (robot must be turned on).



12. Disconnect and use the orange button on the robot to run the program.

Discussion questions

What difficulties did you encounter with the programming?

What did you do to overcome these difficulties?

What strategies did you use to figure out the distance from the obstacle to the X in steps 4-5?

What strategies did you use to figure out how to program the robot to turn in step 6?

Why did you need to select “unlimited” for the move block in Step 2?

You needed to select “unlimited” because you didn’t know how away the obstacle was that would activate the ultrasonic sensor.

What other things might use a touch sensor?

Automatic car wash

Touch screens

What other things might use an ultrasonic (distance) sensor?

Bats use ultrasound for navigation.

Many motion detectors use ultrasound (burglar alarms, motion sensing lights).

Scoring Rubric

Task	**Points Possible**	**Points Earned**
The robot begins the challenge centered on the X facing toward the obstacle (ultrasonic sensor facing obstacle)	**10**	
The robot moves toward the obstacle	**10**	
The robot uses the ultrasonic sensor to detect the obstacle at a distance of 12 inches	**10**	
Upon sensing the obstacle the robot moves backwards	**10**	
The robot stops in its original position (centered on the X as in step 1)	**10**	
The robot spins in place 2 and ½ times such that the touch sensor is now facing the obstacle	**10**	
The robot moves toward the obstacle	**10**	
The robot touches the obstacle with the touch sensor	**10**	
The robot travels back to the X	**10**	
The robot stops when centered on the X	**10**	
Total	**100**	

Instructor Guide
Challenge Title: Magic Touch

Introduction

This robotic challenge uses a challenge mat. The robot configuration must include the sound sensor and the touch sensor. The obstacle listed on the challenge schematic can be a wall or any other object that will not move when struck by the robot. The participants will not be told the distance from the start box to the corner nor will they be given the distance from the corner to the obstacle. They will need to use trial-and-error to calculate the proper travel duration of the robot. For example, they might have to view how far the robot travels in one tire rotation and then estimate how many rotations it will take to cover the required distance. The youth will also have to use trial-and-error to figure out how to make the robot turn the correct arc distance so the touch sensor is facing the obstacle. They may also reference "Distance and Turns" to estimate how many rotations or degrees it will take to successfully complete the challenge.

Objectives

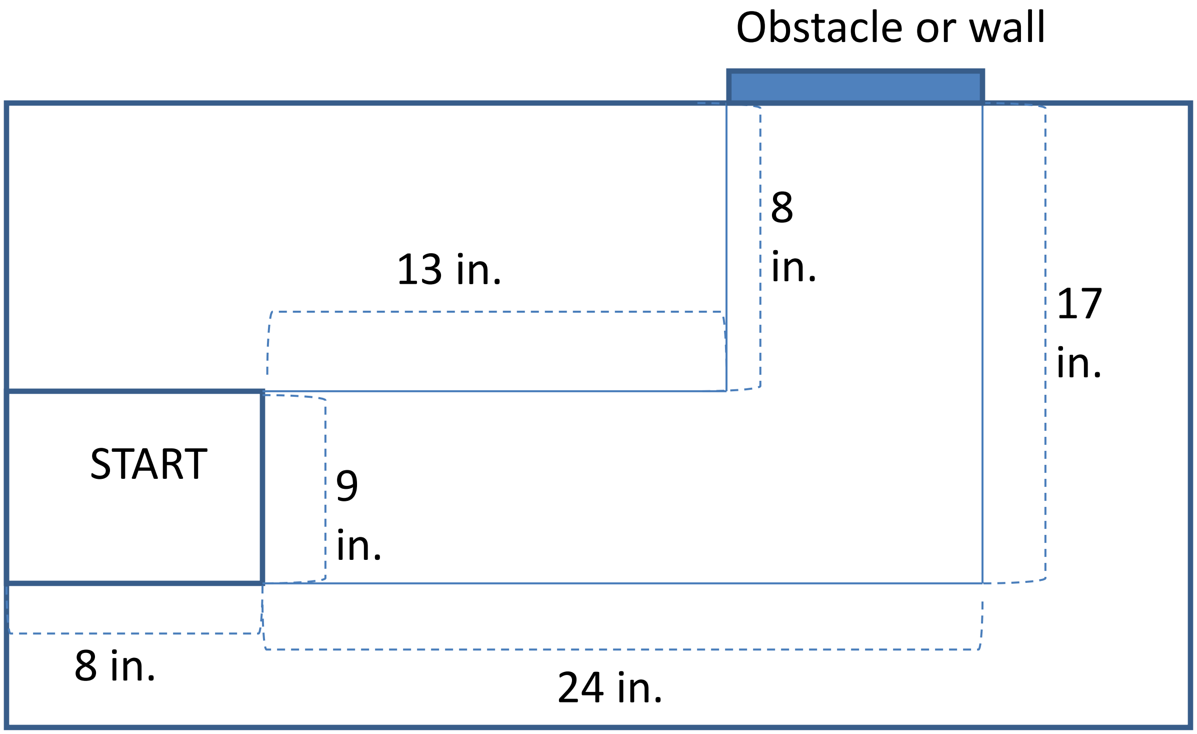
Youth will demonstrate creativity, innovation and critical thinking skills.
Youth will increase their ability to work collaboratively with others.
Youth will be able to improve communication skills.
Youth will better understand the process of programming and evaluating robotic movements.
Youth will increase their ability to design a solution to a challenge.

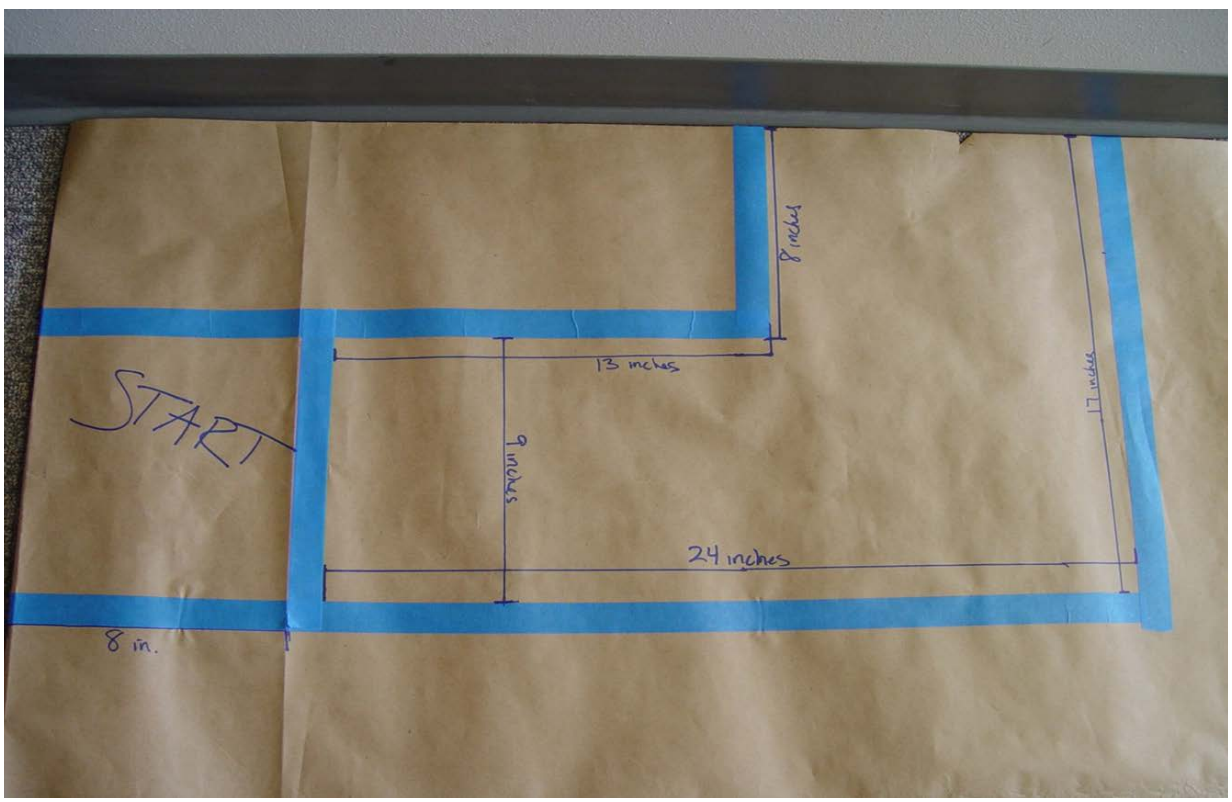
Level of Difficulty ☆☆☆

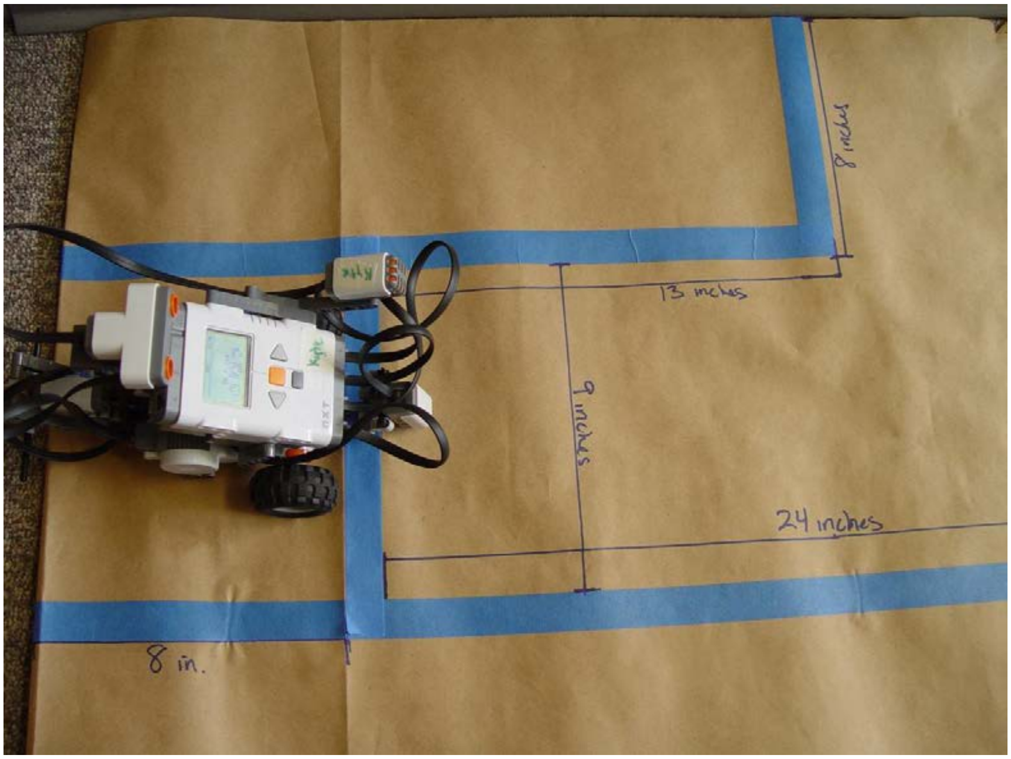
Preparation and Materials

Instructor should determine the teaching methodology (as identified in the introduction of this guide) that best fits the audience.
Print appropriate student materials.
Create "Magic Touch" challenge mat using:

- Masking or painter's tape
- Measuring tape
- Paper or poster board (at least 40 X 20 inches in size)
- Obstacle such as a weighted box, heavy book or wall


Obstacle or wall
13 in.
8 in.
17 in.
START
9 in.
8 in.
24 in.


START
8 inches
13 inches
9 inches
17 inches
24 inches
8 in.



This picture demonstrates the position of the robot in step 1 and at the end of step 6

Time Required
Programming: 30-45 minutes (if participants are given text instructions only)
To run the course: 1-2 minutes

Procedure of Programming Steps
Depending on the expertise level of the participants, the instructor can give the youth the text of the steps involved in the challenge, give them the abbreviated programming guide or give them the entire pictorial programming guide.

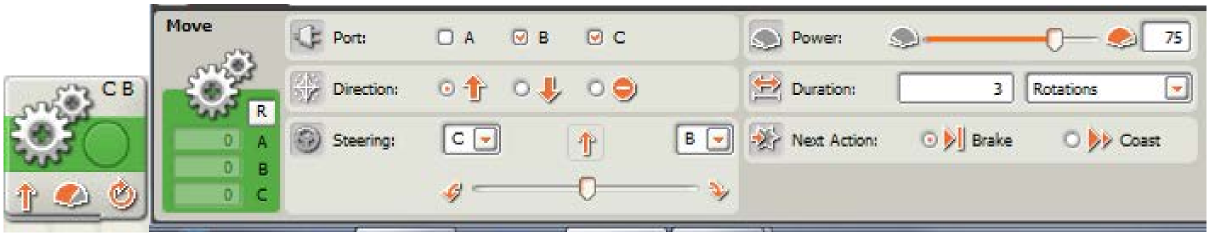
TEXT:

1. The front wheels of the robot must be completely behind the start line.
2. The robot must stay inside the lines at all times.
3. The robot moves from the start area to the corner and stops.
4. The robot turns when it hears a sound, such that the touch sensor is facing the obstacle.
5. The robot touches the obstacle with the touch sensor.
6. Once the robot touches the wall it must return to its initial starting position (facing forward and ready to complete the course again).

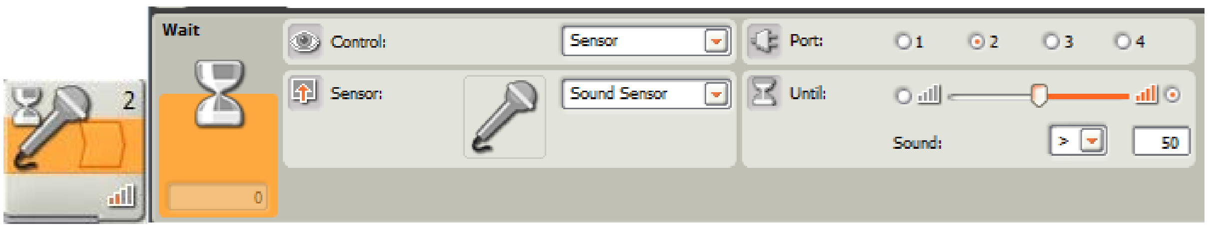
ABBREVIATED PROGRAMMING GUIDE:

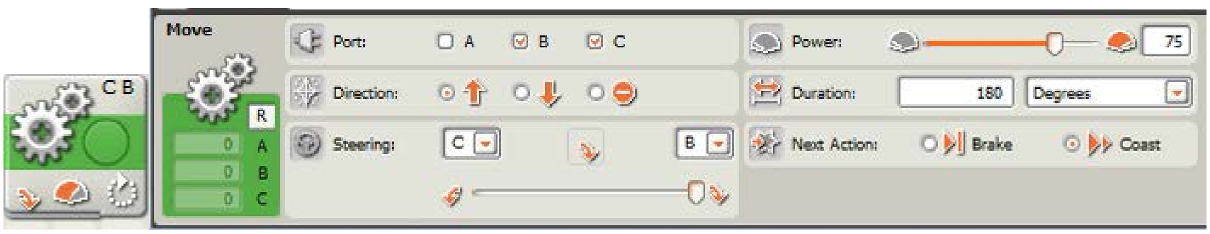
COMPLETE PICTORIAL PROGRAMMING GUIDE:

1. The front wheels of the robot must be completely behind the start line.
2. The robot must stay inside the lines at all times.
3. The robot moves from the start area to the corner and stops.

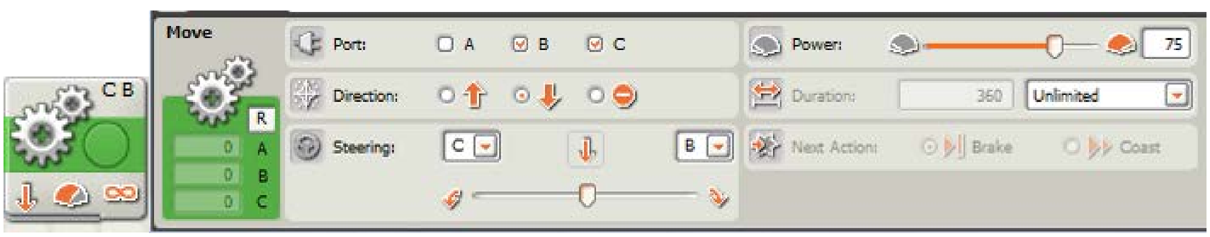


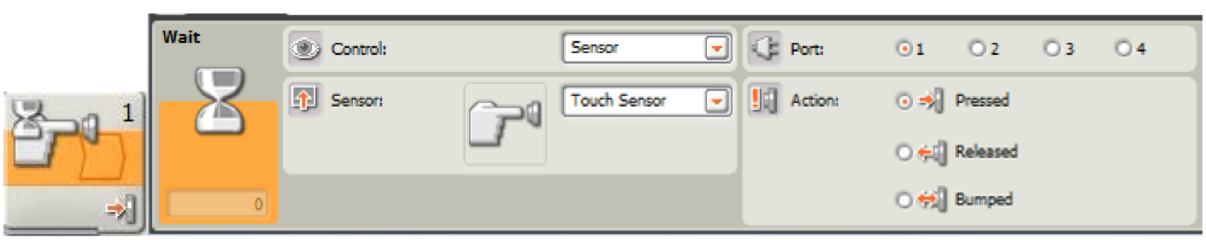
4. The robot turns when it hears a sound, such that the touch sensor is facing the obstacle.



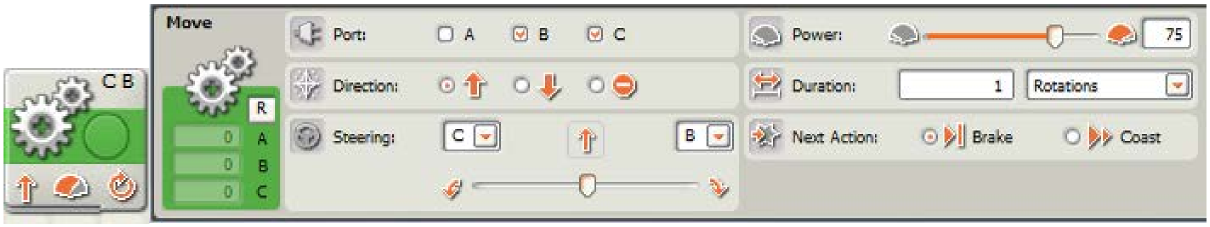


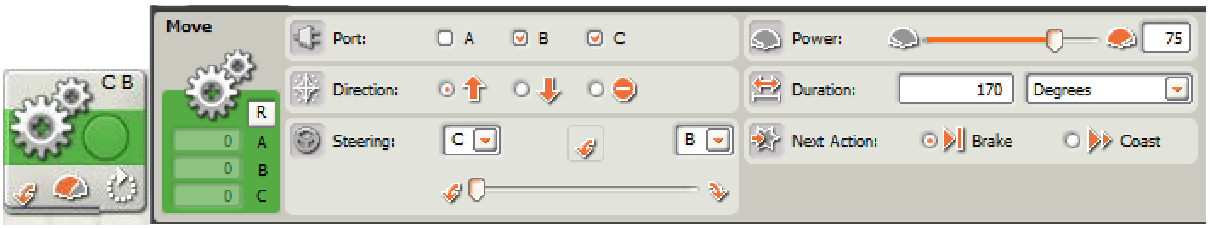
5. The robot touches the obstacle with the touch sensor.

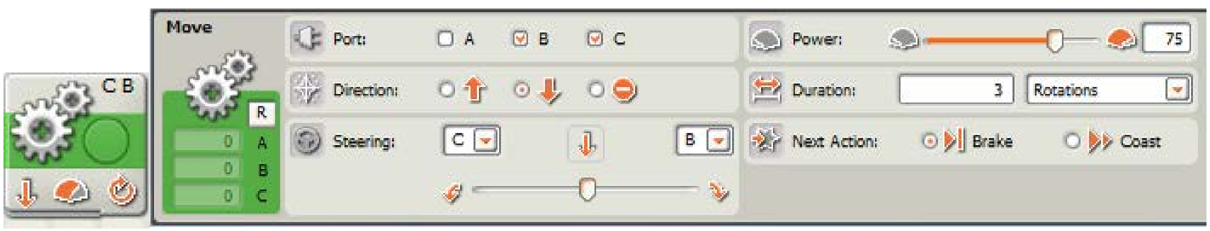




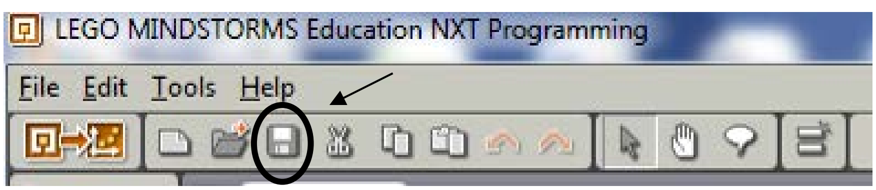
6. Once the robot touches the wall it must return to its initial starting position (facing forward and ready to complete the course again).



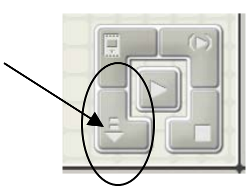




7. Save the computer program.



8. Download the program onto the robot using the USB cord (robot must be turned on).



9. Disconnect and use the orange button on the robot to run the program.

Discussion Questions
What difficulties did you encounter with the programming?

What did you do to overcome these difficulties?

What strategies did you use to figure out the distance from the start area to the turn corner?

What strategies did you use to figure out how to program the robot to turn in step 4?

Why did you need to select “unlimited” for the move block in Step 5?

You needed to select “unlimited” because you didn’t know how away the obstacle was that would activate the touch sensor.

What other things might use a touch sensor?

Automatic car wash

Touch screens

What other things might use a sound sensor?

Any voice activation software: Siri, etc.

Many new car radios will decrease sound when engine noise is reduced.

Scoring Rubric

Action	Points Possible	Points Earned
The front wheels of the robot must be completely behind the start line	10	
The robot must stay inside the lines at all times	10	
The robot moves from the start area to the corner and stops	10	
The robot turns when it hears a sound, such that the touch sensor is facing the obstacle	10	
The robot touches the obstacle with the touch sensor	10	
Once the robot touches the wall it must return to its initial starting position (facing forward and ready to complete the course again)	10	
Total	60	

Instructor Guide
Challenge Title: Cliff Hanger

Introduction

This robotic challenge uses a challenge mat. The robot configuration must include the light sensor which is placed on the front of the robot with the sensor pointing down as describes on pages 32-34 of the 9797 building guide (basic robot configuration). The light sensor will detect the change in color from a lighter background to a darker background. This means in steps 4 and 7 the light sensor must have the left radio button selected indicating the robot is sensing something darker than the background. The dark paper signifies a cliff that the robots wheels must not touch during the challenge. The paper used in the schematic is a sheet of black construction paper cut in half widthwise and taped to the background paper.

Objectives

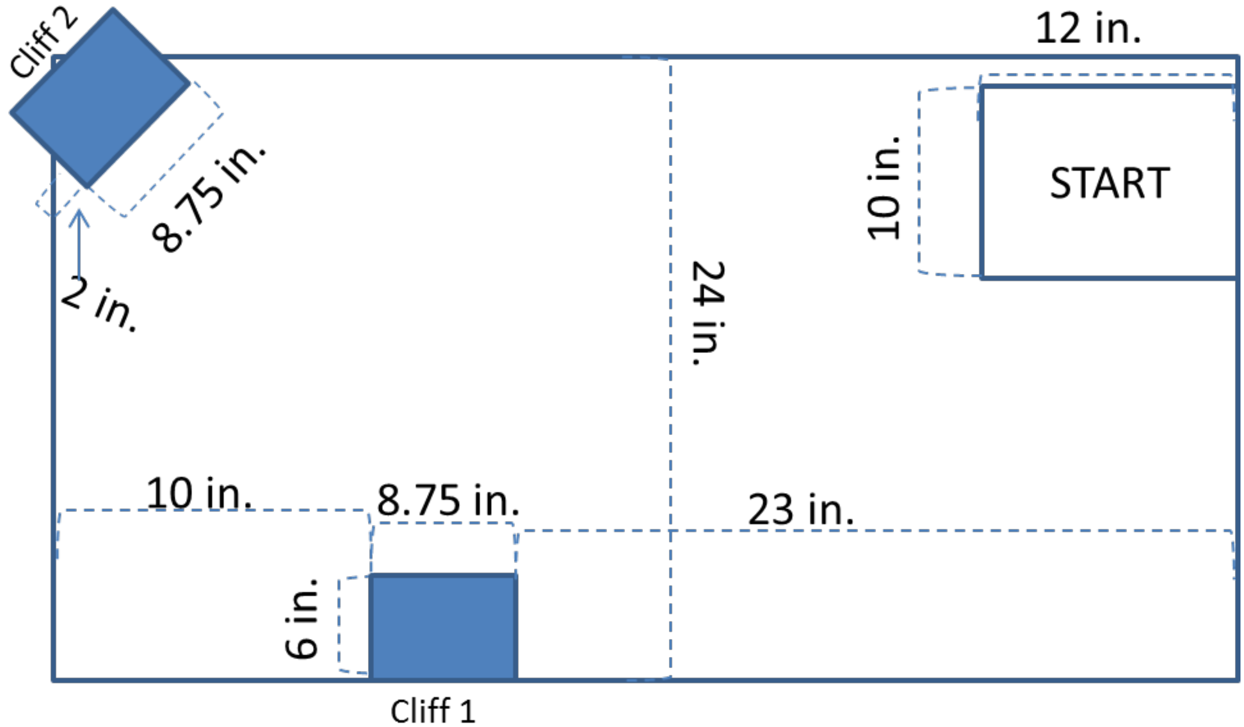
Youth will demonstrate creativity, innovation and critical thinking skills.
Youth will increase their ability to work collaboratively with others.
Youth will be able to improve communication skills.
Youth will better understand the process of programming and evaluating robotic movements.
Youth will increase their ability to design a solution to a challenge.

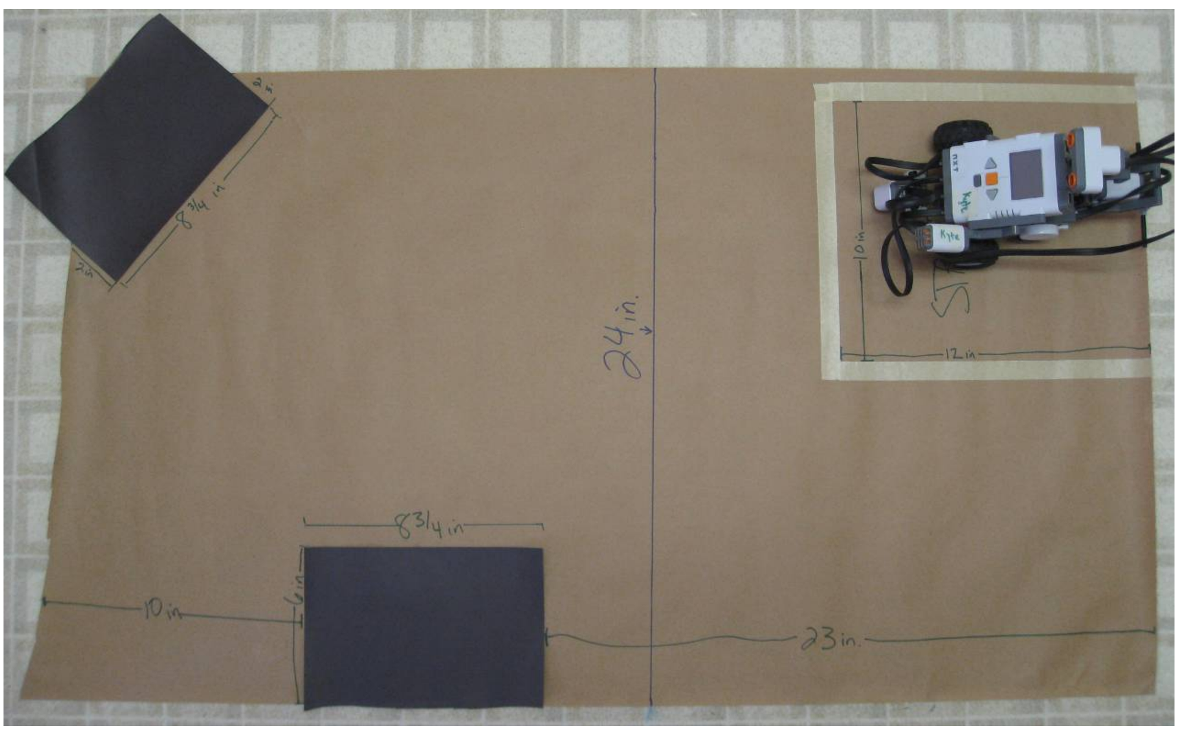
Level of Difficulty ☆☆☆☆

Preparation and Materials

Instructor should determine teaching methodology (as identified in the introduction of this guide) that best fits the audience.
Print appropriate student materials.
Create "Cliff Hanger" challenge mat using:

- Masking or painter's tape
- Measuring tape
- Paper or poster board of a light or neutral color (at least 45 X 24 inches in size)
- Black paper or tape





This picture demonstrates the position of the robot in step 1 and step 8.

Time Required
Programming: 45 minutes-1 hour (if participants are given text instructions only)
To run the course: 1-2 minutes

Procedure of Programming Steps
Depending on the expertise level of the participants, the instructor can give the youth the text of the steps involved in the challenge, give them the abbreviated programming guide or give them the entire pictorial programming guide.

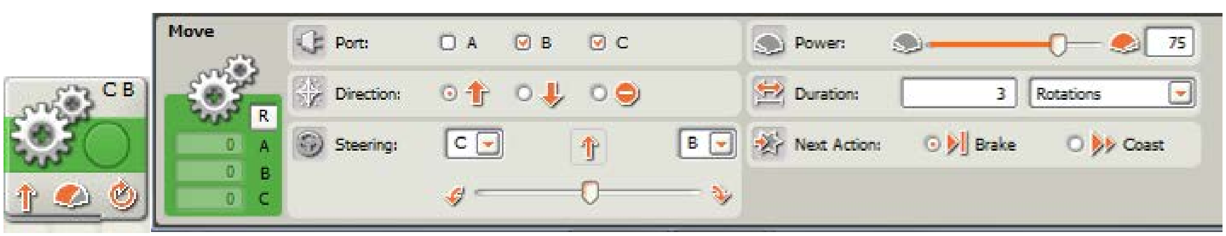
TEXT:

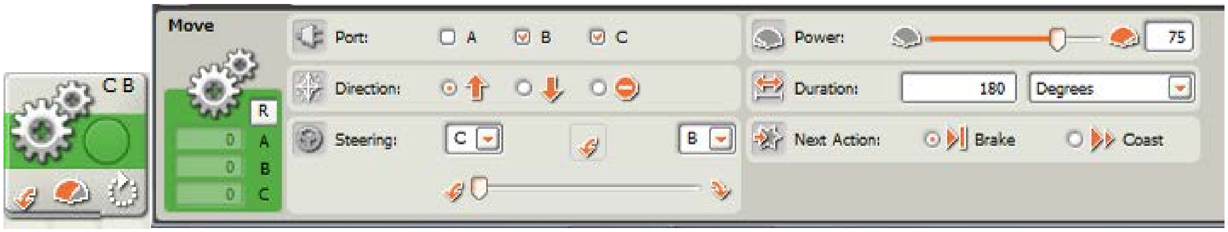
1. The robot must begin the challenge completely inside the start box.
2. The robot must stay on the challenge mat at all times during the challenge.
3. The robot moves forward toward cliff 1.
4. The robot senses cliff 1 with the light sensor, stops, and says “watch out”.
5. The robot backs away from cliff 1.
6. The robot moves in the forward direction toward cliff 2.
7. The robot senses cliff 1 with the light sensor, stops, and says “ahnoo”.
8. The robot moves backwards to the start box and completes the course in the same position as step 1.

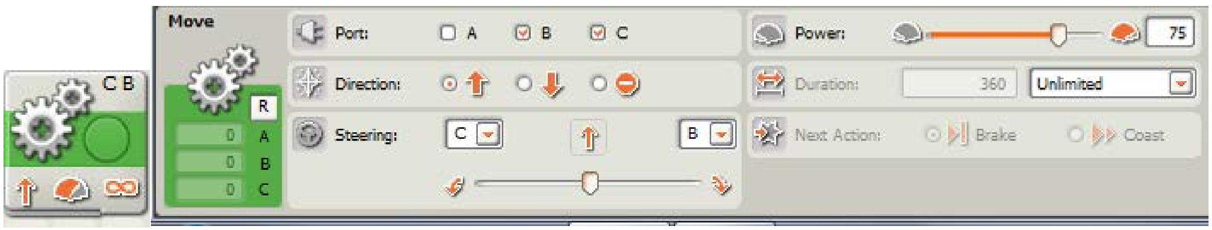
ABBREVIATED PROGRAMMING GUIDE:

COMPLETE PICTORIAL PROGRAMMING GUIDE

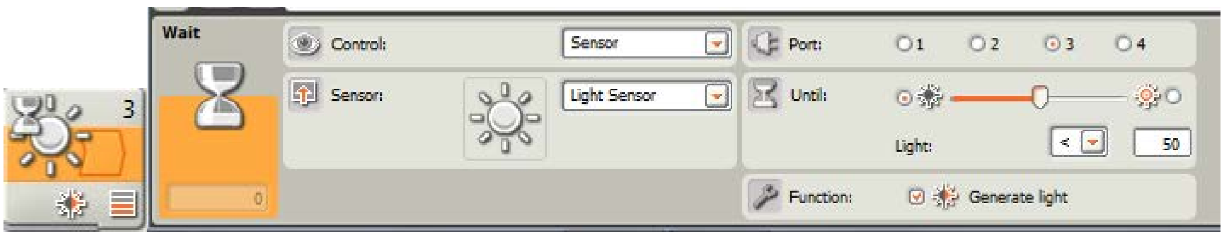
1. The robot must begin the challenge completely inside the start box.
2. The robot must stay on the challenge mat at all times during the challenge.
3. The robot moves forward toward cliff 1.

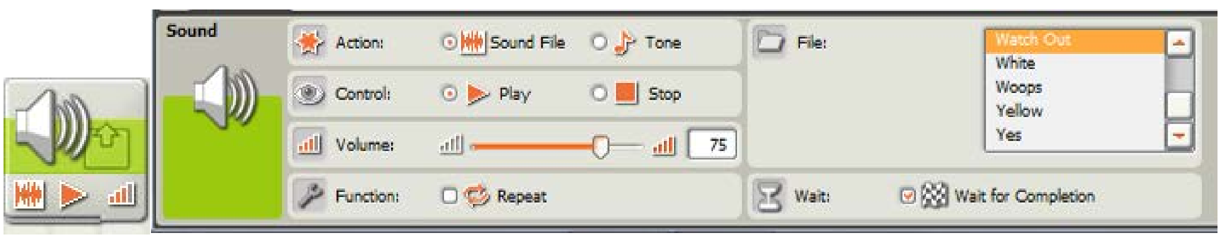




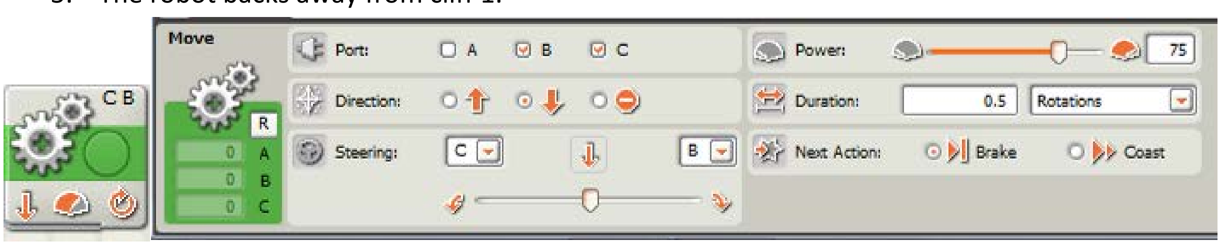


4. The robot senses cliff 1 with the light sensor, stops, and says "watch out".

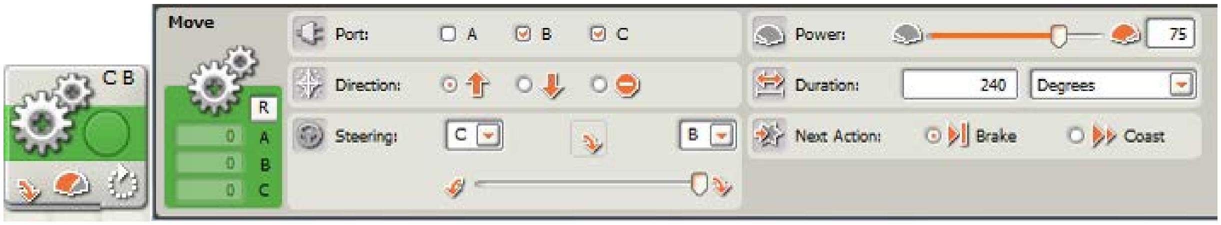


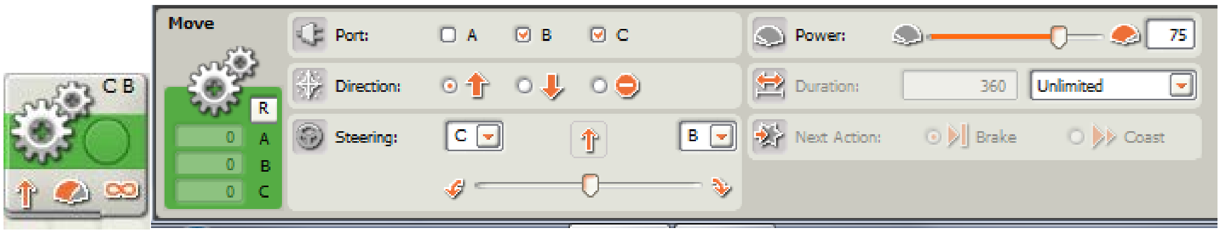


5. The robot backs away from cliff 1.

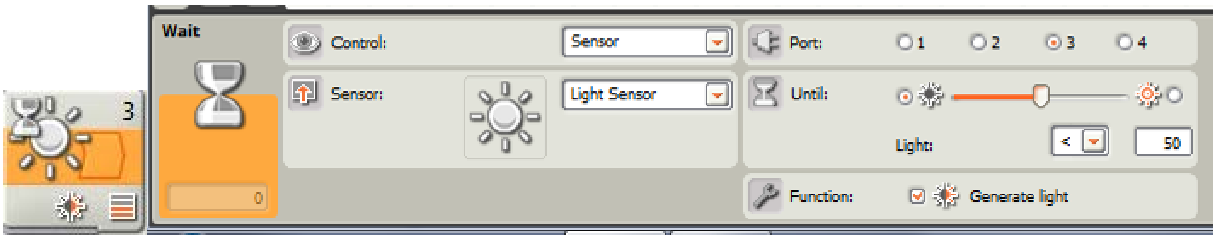


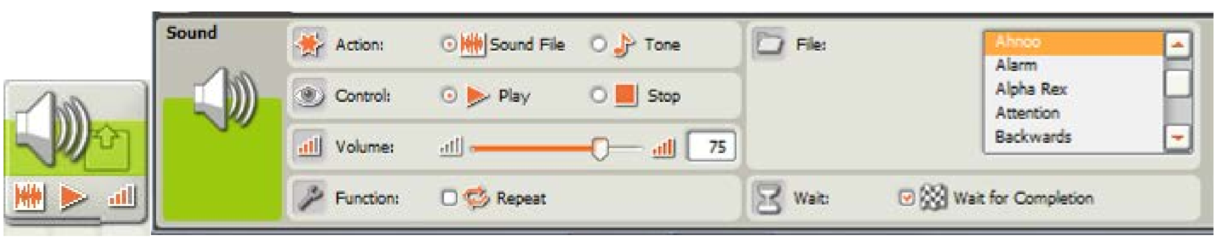
6. The robot moves in the forward direction toward cliff 2.



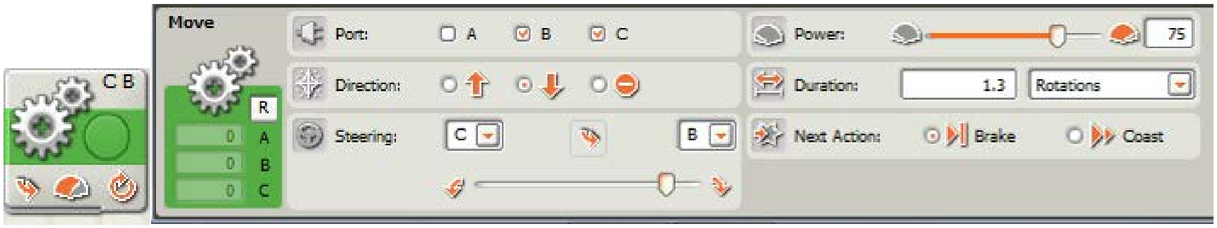


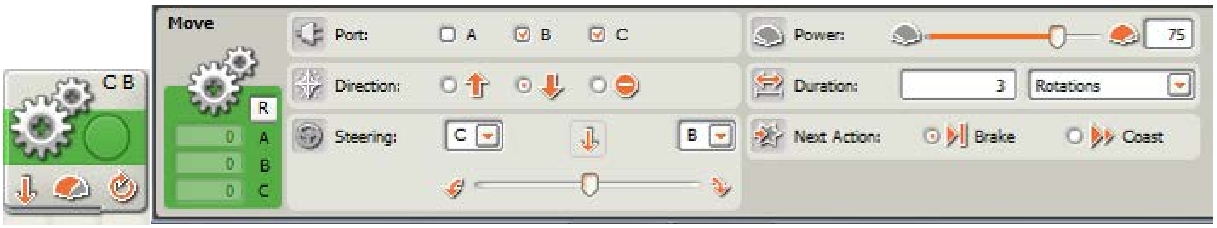
7. The robot senses cliff 1 with the light sensor, stops, and says “ahnoo”.



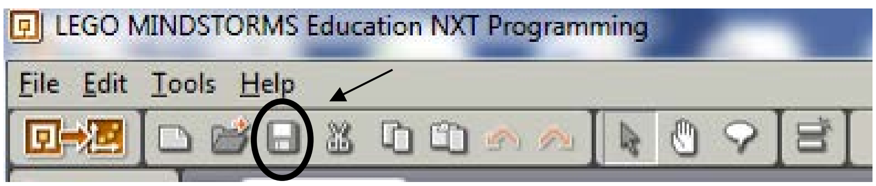


8. The robot moves backwards to the start box and completes the course in the same position as step 1.

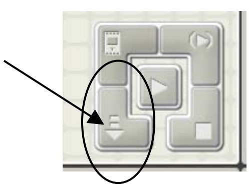




9. Save the computer program.



10. Download the program onto the robot using the USB cord (robot must be turned on).



11. Disconnect and use the orange button on the robot to run the program.

Discussion Questions

What difficulties did you encounter with the programming?

What did you do to overcome these difficulties?

What strategies did you use to figure out the distance from cliff 2 to the start box in step 8?

Why did you have to select the left radio button in the light sensor programming to detect the cliffs?

The robot was moving from a lighter background to a darker background. If the right radio button was selected the robot would not be able to detect the black cliff paper.

Why did you need to select “unlimited” for the move block in Step 3?

You needed to select "unlimited" because you wanted the robot to move until the light sensor was activated. You did not know the distance from the start box to the cliff.

What other things might use a light sensor?

Many garage doors use a light sensor. Two sensors are placed on either side of the door, when something breaks the light beam the door automatically stops.

Most electronics use a light sensor to automatically adjust the screen brightness.

Scoring Rubric

Action	**Points Possible**	**Points Earned**
The robot must complete the challenge completely inside the start box	**10**	
The robot must stay on the challenge mat at all times during the challenge.	**10**	
The robot moves forward toward cliff 1	**10**	
The robot senses cliff 1 with the light sensor, stops	**10**	
The robot says "watch out"	**10**	
The robot backs away from cliff 1	**10**	
The robot moves in the forward direction toward cliff 2	**10**	
The robot senses cliff 1 with the light sensor, stops, and says "ahnoo"	**10**	
The robot moves backwards to the start box	**10**	
The robot and completes the course in the same position as step 1	**10**	
Total	**100**	

Instructor Guide

Challenge Title: Breakfast Bot (Modified from 2011 Wisconsin 4-H Robotics Rally Implementation Packet) Original author: Judy Wolniakowski (retired), Brown County 4-H Youth Development.

Introduction

This robotic challenge uses a challenge mat. The robot configuration must include the ultrasonic (distance) sensor and the touch sensor. Other sensors may be used throughout the routine to complete the tasks outlined in the challenge. Since there are several ways to successfully complete the challenge, no abbreviated or pictorial programming guide is provided. To successfully complete the Breakfast Bot challenge, the participant must detect and avoid several objects placed on the periphery of the challenge mat.

Objectives

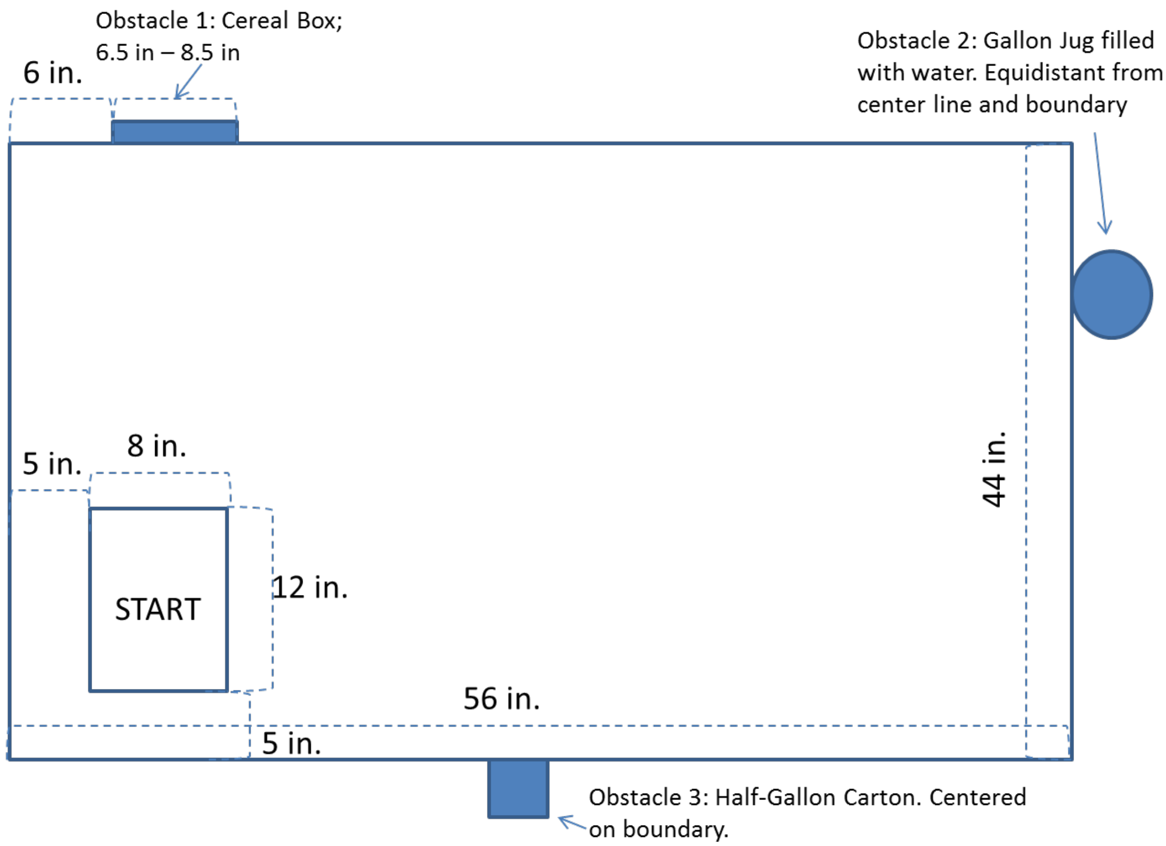
Youth will demonstrate creativity, innovation and critical thinking skills.
Youth will increase their ability to work collaboratively with others.
Youth will be able to improve communication skills.
Youth will better understand the process of programming and evaluating robotic movements.
Youth will increase their ability to design a solution to a challenge.

Level of Difficulty ☆☆☆☆☆

Preparation and Materials

Instructor should determine the teaching strategy (as defined in the introduction of this guide) that best fits the audience.
Print appropriate student materials.
Create "Breakfast Bot" challenge mat using:

- Measuring tape
- Masking or painter's tape
- Paper or poster board (at least 56 X 44 inches in size)
- 1 cereal box (6.5 in. – 8.5 in. in width)
- 1 gallon jug (weighted with water)
- 1 half-gallon carton (weighted with water)


Obstacle 1: Cereal Box;
6.5 in – 8.5 in
6 in.
Obstacle 2: Gallon Jug filled
with water. Equidistant from
center line and boundary
5 in.
8 in.
44 in.
START
12 in.
56 in.
5 in.
Obstacle 3: Half-Gallon Carton. Centered
on boundary.

Time Required
Programming: 3 hours or more
To run the course: 2-3 minutes

Procedure of Programming Steps
Since there are several ways to successfully complete the challenge, no abbreviated or pictorial programming guide is provided. This challenge is meant for advanced youth and will enable them to practice critical thinking, creativity and robotic programming mastery skills.

TEXT:
1. The robot begins the challenge completely within the start box.
2. The entire robot must stay on the challenge field at all times during the routine.
3. The robot moves forward to detect obstacle 1 with the ultrasonic sensor.
4. The robot will travel backwards where it will use the touch sensor to detect obstacle 2.
5. The robot will move forward to the center of the playing field.
6. The robot will turn to face obstacle 3 (ultrasonic sensor facing obstacle).
7. The robot will spin in place such that the touch sensor is facing the obstacle.
8. The robot will detect obstacle 3 with by hitting it with the touch sensor.
9. The robot will finish the routine entirely within the start box facing the same direction as in step 1.

Discussion Questions
What difficulties did you encounter with the programming?

What did you do to overcome these difficulties?

What strategies did you use to program your robot to spin in step 7?

What other things might use a touch sensor?

Automatic car wash
Touch screens

What other things might use an ultrasonic (distance) sensor?

Bats use ultrasound for navigation
Many motion detectors use ultrasound (burglar alarms, motion sensing lights)

Scoring Rubric

Action	Points Possible	Points Earned
The robot begins the challenge completely within the start box.	**10**	
The entire robot must stay on the challenge field at all times during the routine.	**10**	
The robot moves forward to detect obstacle 1 with the ultrasonic sensor.	**10**	
The robot will travel backwards where it will use the touch sensor to detect obstacle 2.	**10**	
The robot will move forward to the center of the playing field	**10**	
The robot will turn to face obstacle 3 (ultrasonic sensor/front facing obstacle).	**10**	
The robot will spin in place such that the touch sensor/back is facing the obstacle.	**10**	
The robot will detect obstacle 3 with by hitting it with the touch sensor	**10**	
The robot will finish the routine entirely within the start box	**10**	
The robot will facing the same direction as it started	**10**	
Total	**10**	

Instructor Guide

Challenge Title: GOAL!! (Modified from 2011 Wisconsin 4-H Robotics Rally Implementation Packet)

Original author: Michelle Grimm, Taylor County 4-H Youth Development

Introduction

This robotic challenge uses a challenge mat. The robot configuration must include the sound sensor. Other sensors may be used throughout the routine to complete the tasks outlined in the challenge. Since there are several ways to successfully complete the challenge, no abbreviated or pictorial programming guide is provided. To successfully complete the GOAL! challenge, the participant must score a goal by knocking the ball off the platform sometime during the program routine. The ball is placed in the middle of the challenge mat by balancing it on three connected Legos as shown in the schematics below. Youth may build the rotating arm, if they wish; however, they may also knock the ball off the platform using any part of the robot. The robot must also touch all corner lines of the field. Both front tires should touch and/or cross the corner lines to exhibit mastery of this challenge.

Objectives

Youth will demonstrate creativity, innovation and critical thinking skills.

Youth will increase their ability to work collaboratively with others.

Youth will be able to improve communication skills.

Youth will better understand the process of programming and evaluating robotic movements.

Youth will increase their ability to design a solution to a challenge.

Level of Difficulty ☆☆☆☆☆

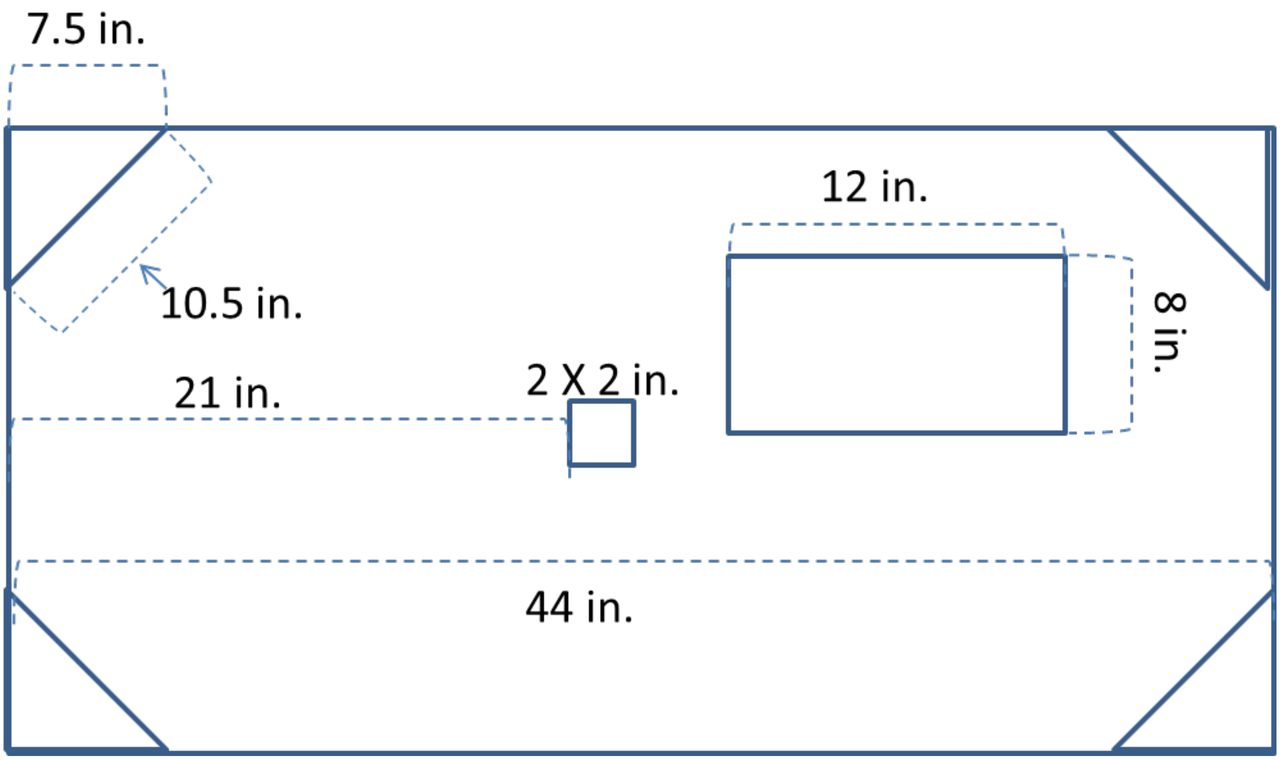
Preparation and Materials

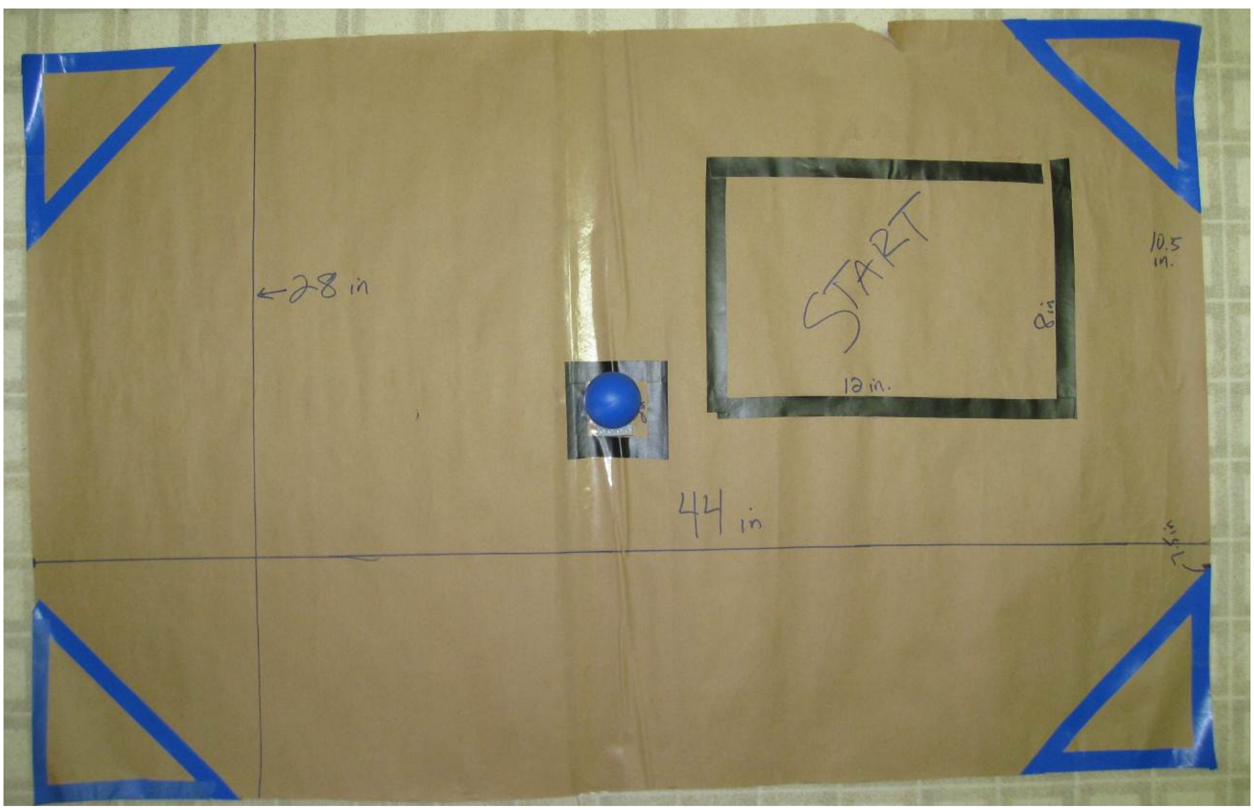
Instructor should determine the teaching methodology (as identified in the introduction of this guide) that best fits the audience.

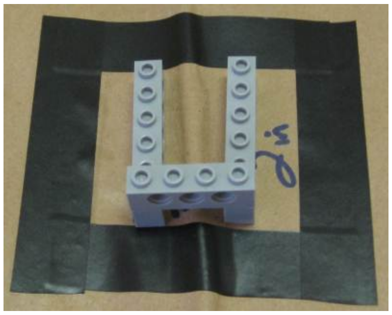
Print appropriate student materials.

Create “GOAL!!” challenge mat using:

- Masking or painter’s tape
- Measuring tape
- Paper or poster board (at least 44 X 28 inches in size)
- Lego plastic ball (as provided in the LEGO® Mindstorms® NXT v. 2.0 Kit)
- Two 1X8 LEGO bricks
- One 1X6 LEGO brick


7.5 in.
12 in.
10.5 in.
8 in.
2 X 2 in.
21 in.
44 in.


28 in
START
10.5 in.
12 in.
44 in



Ball Platform

Time Required
Programming: 3 hours or more
To run the course: 2-3 minutes

Procedure of Programming Steps
Since there are several ways to successfully complete the challenge, no abbreviated or pictorial programming guide is provided. This challenge is meant for advanced youth and will enable them to practice critical thinking, creativity and robotic programming mastery skills.

TEXT:
1. The robot begins the challenge completely within the start box.
2. The entire robot must stay on the challenge field at all times during the routine.
3. The robot must be set in motion with the clap of a hand.
4. The robot must now proceed to access each of the corners of the field. The robot must have two wheels within the boundary of each corner of the field without leaving the field.
5. The corners can be accessed in any order but all four must be crossed.
6. At some time during the routine, the robot must approach the ball stand placed at the center square and score a goal by knocking the ball off the platform. The ball does not need to leave the challenge field, just be removed from the platform.
7. At the end of the routine the robot must return to the start box and must announce in some way that it has completed the task. The robot must be entirely within the start box.

Discussion Questions
What difficulties did you encounter with the programming?

What did you do to overcome these difficulties?

Why might you want to wait until you have completed touching all the corners before hitting the ball?

It would be hard to predict where the ball would roll to and the position of the ball might interfere with your routine.

Scoring Rubric

Action	Points Possible	Points Earned
The robot begins the challenge completely within the start box	**10**	
The entire robot must stay on the challenge field at all times during the routine	**10**	
The robot must be set in motion with the clap of a hand	**10**	
The robot must now proceed to access each of the corners of the field. The robot must have two wheels within the boundary of each corner of the field without leaving the field. The corners can be accessed in any order but all four must be crossed		
Corner 1	**10**	
Corner 2	**10**	
Corner 3	**10**	
Corner 4	**10**	
At some time during the routine, the robot must approach the ball stand placed at the center square and score a goal by knocking the ball off the platform. The ball does not need to leave the challenge field, just be removed from the platform	**10**	
At the end of the routine the robot must return to the start box and must announce in some way that it has completed the task.	**10**	
The robot must be entirely within the start box	**10**	
Total	**100**	

Robotics videos:

Segment 1-Introduction to NXT Mindstorm Robots: http://youtu.be/0AIHQS2KfTU

Segment 2- Introduction to the NXT software: http://www.youtube.com/watch?v=ngNm1SnaOJA

Segment 3-Record, play, sound and display: http://youtu.be/8mq2iMJOVZw

Segment 4- Move block: http://youtu.be/BgteOkiDjuQ

Segment 5- Ultrasonic sensor: http://youtu.be/4isOg2lnaJY

Segment 6-Touch sensor: http://youtu.be/7ZmOeZhqoEM

Segment 7-Sound sensor: http://youtu.be/qh1xV4OgHpM

Segment 8-Light sensor: http://youtu.be/QxfkyXWFo-0