**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:**

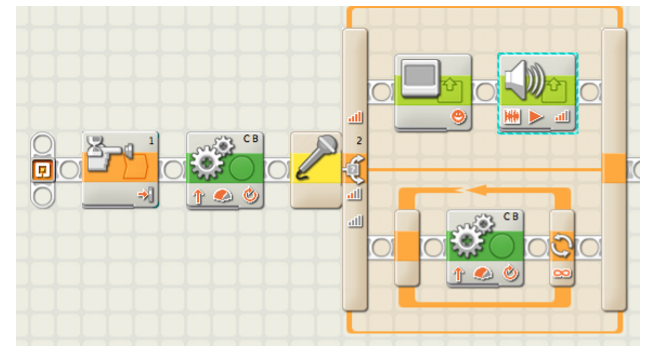
* Review of Program activity from Day 7 (20 minutes)
* Interface: the parts of the Mindstorms NXT software (10 minutes)
* A simple program from the common palette (30 minutes)
* A simple program from the complete palette (40 minutes)
* How to use the tutorials (10 minutes)

**Student Activities:**

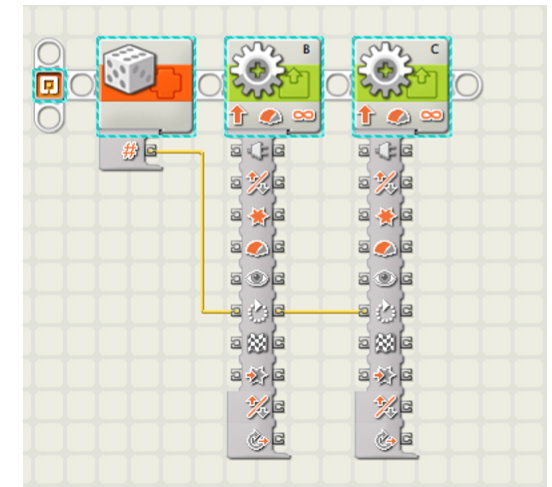
* Discuss how the programs were created in the NXT brick and how they behaved compared to expectations.
* Listen to explanation of Mindstorms NXT software and respond to questions.
* Give ideas to teacher as s/he writes small programs in the software.
* Listen to explanation of how to use the tutorials.

**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