

Programming with the NXT using the touch and ultrasonic sensors

To be used with the activity of the 'How do Human Sensors Work?' lesson

How do you incorporate sensors into your programs?

▶ **What you know:**

- ▶ You can start your own program using the NXT MINSTORMS programming software
- ▶ You can use strings of move commands to navigate your robot through any path required
- ▶ You can make your robot travel for any amount of time or any distance (measured in rotations of the robot's wheels) required
- ▶ (if you do not know how to do these, refer to the activity of the What is a Robot? Lesson)

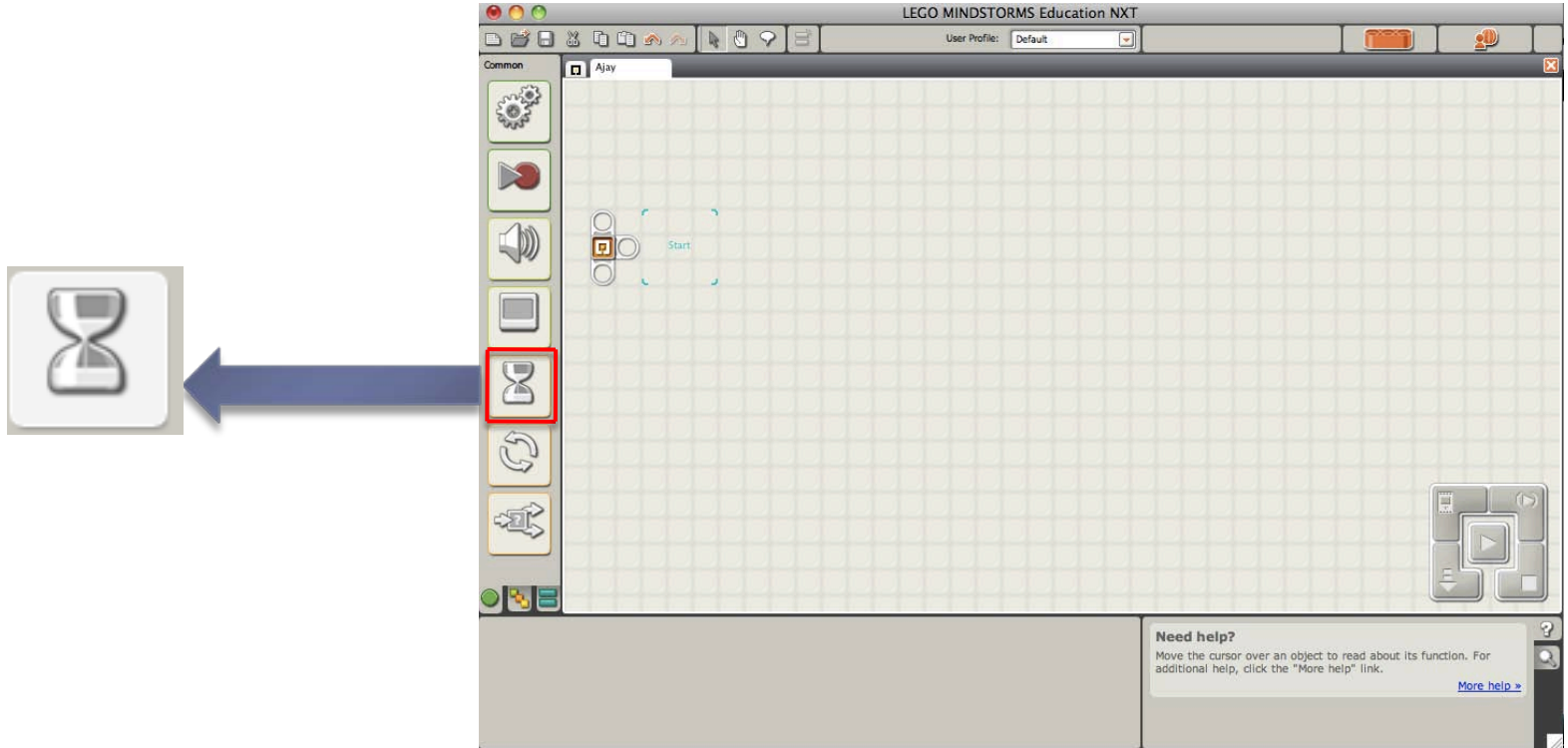
▶ **What you will learn**

- ▶ How to use sensors to aid the navigation of your robot.
-



The “Wait for” command

- ▶ First, we will learn about the ‘wait for command’. This is the icon that is marked with an hourglass on the left side of your programming screen.



Wait for a sensor...contd.

- ▶ **When you scroll over the “wait for” icon, several options will come up.**



- ▶ **Since we are concerned only with moving a robot around, we will only focus on the second and last options, the ‘wait for touch’ and the ‘wait for distance’ options**



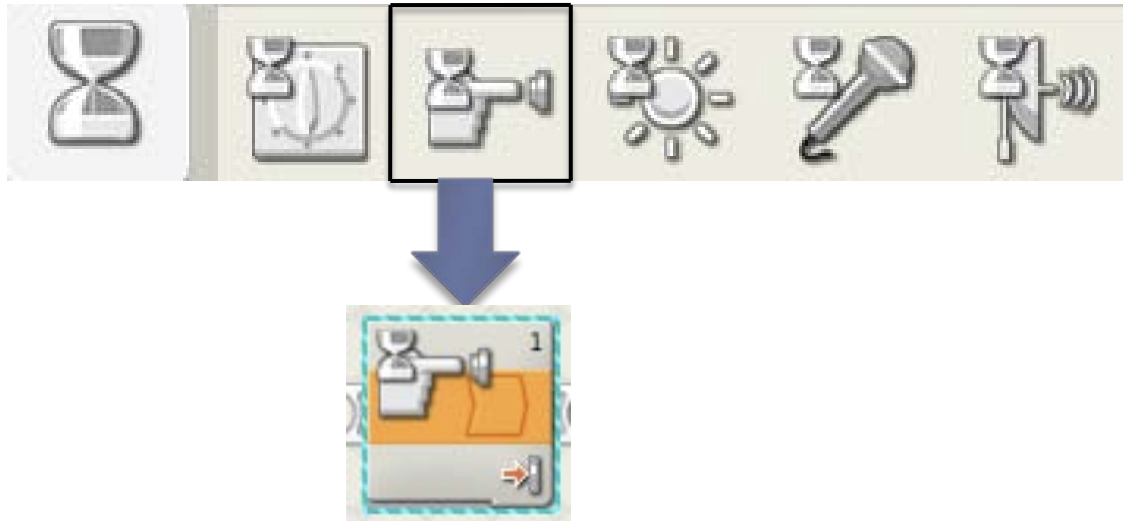
Wait for a sensor...contd.

- ▶ **The first thing to do before using one of the two 'wait for' commands for this lesson is to have a move command that has an unlimited duration.**
- ▶ **This means that the robot will continue to follow the unlimited move command until your sensor is activated**



Wait for touch

- ▶ The second icon is the “wait for touch” icon.

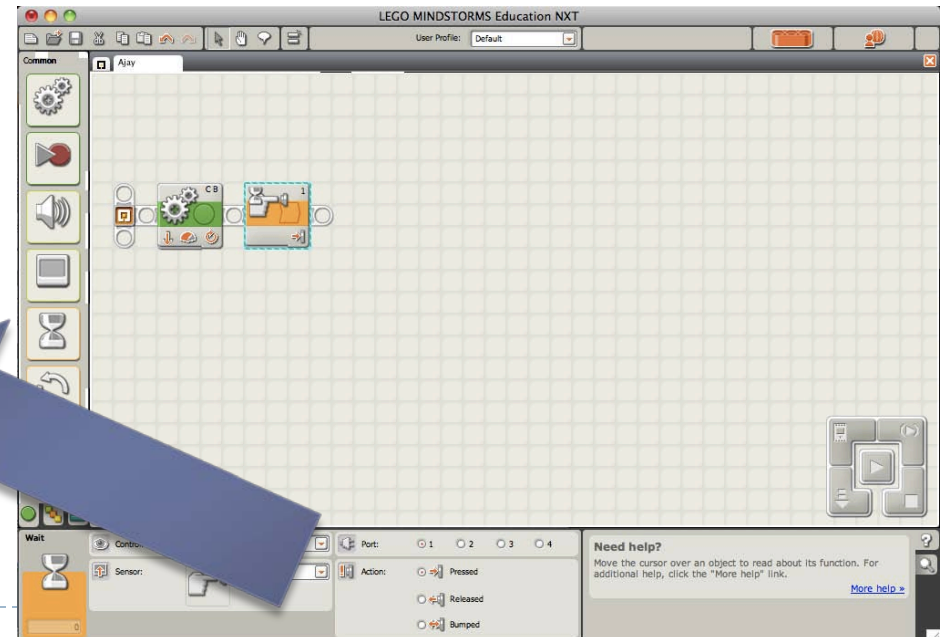
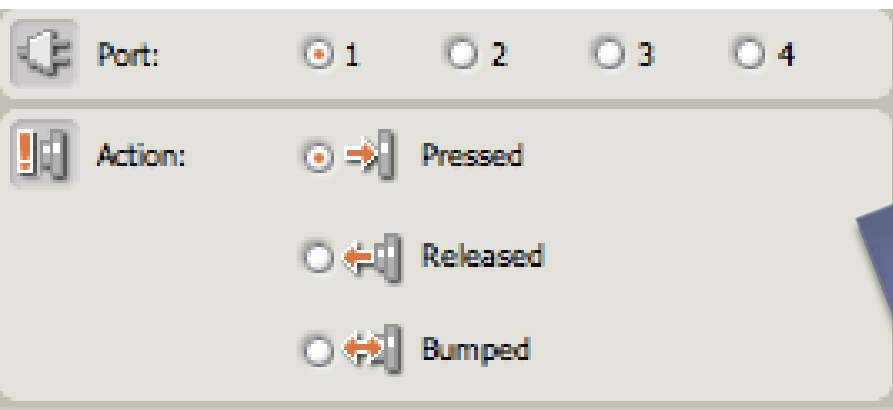


You can use this command to change the behavior of your robot after the touch sensor has been pressed.



Wait for touch – contd.

- ▶ When you click on your ‘wait for touch’ command, a set of options comes up at the bottom of the screen. Make sure that the port selected matches the port that you have plugged your touch sensor into. Then, make sure that the ‘pressed’ option is selected, *not bumped or released!*



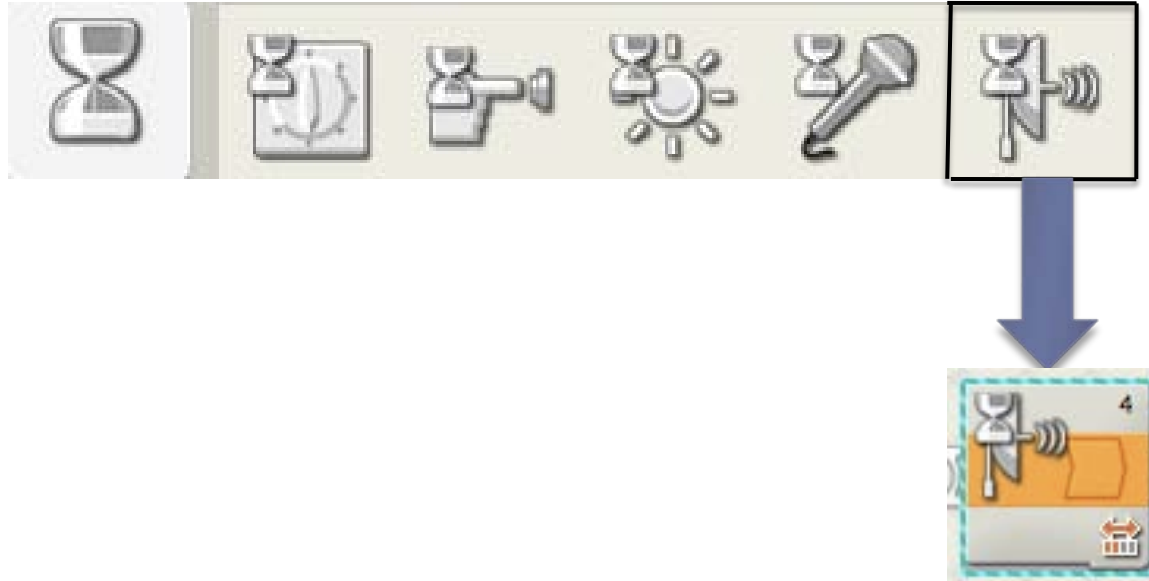
Wait for touch – contd.

- ▶ **Added note: if you have assembled the touch sensor using the instructions given with this lesson, then your touch sensor is on the *back* of the robot.**
- ▶ This means that any unlimited command placed before your 'wait for touch' must be moving in the backwards direction in order for the touch sensor to be activated by the robot bumping into something!



Wait for distance

- ▶ The last icon is the 'wait for distance' icon

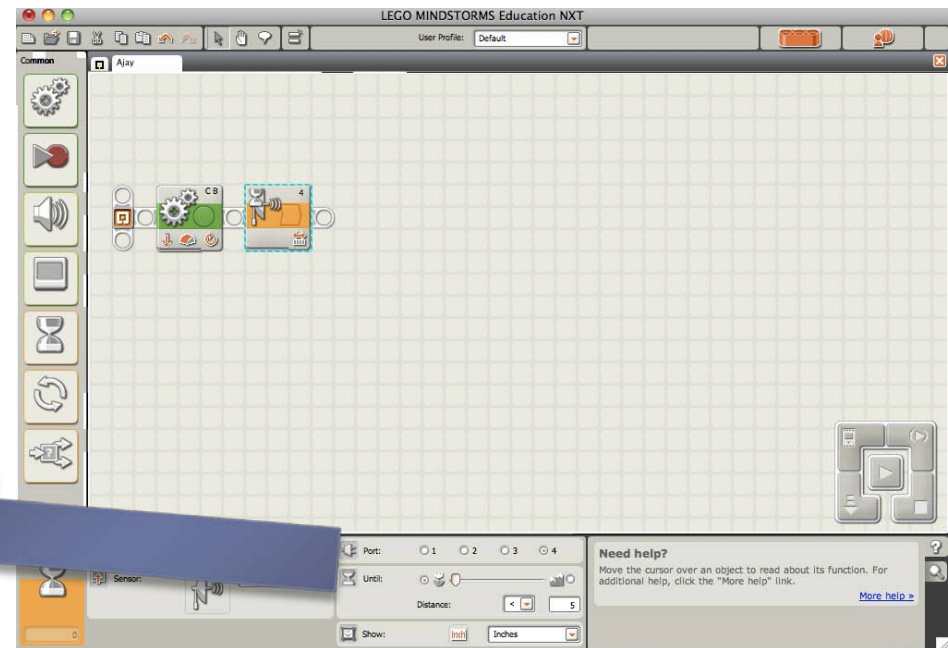


- ▶ You can use this icon to change the behavior of your robot once it detects something with its ultrasonic sensor
-



Wait for distance – contd.

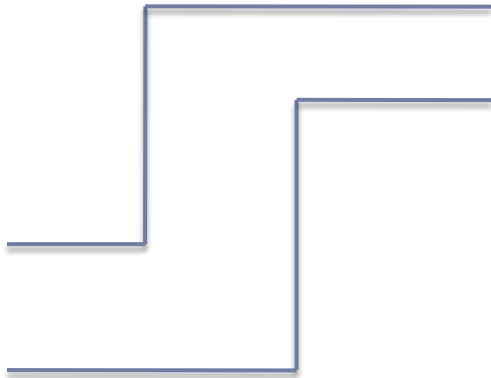
- ▶ When you click on the ‘wait for distance’ command, again a set of options comes up. Again, ensure that the port selected matches the port that you have plugged your sensor into.



A good distance to choose is about 12 inches (make sure you don't use centimeters without meaning to!)

Sample Maze and programs

- ▶ Here is a sample maze



Note: all unlimited straight move commands are going backwards since we are using a touch sensor!!

- ▶ Now, a sample program to follow this maze using a touch sensor:

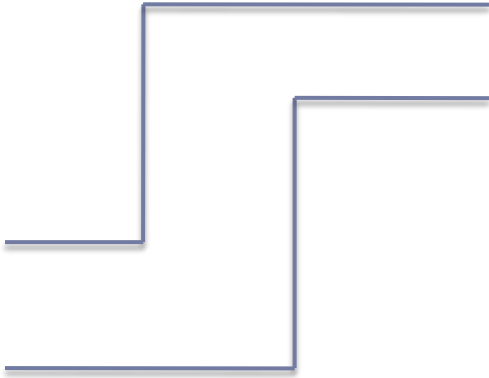


Also note: the turns are done exactly how they were without any

- ▶ **sensor at all! Just pull the bar all the way over and use 0.5 rotations!**

Sample Maze and programs – contd.

- ▶ Using the ultrasonic sensor:



The only difference between using the touch and ultrasonic sensors is that the robot moves *forward* with the ultrasonic sensor and *backward* with the touch sensor!!

Maze tips

- ▶ **The key is the for each straight portion of the maze you need**
 - ▶ An unlimited move command making the robot go straight
 - ▶ Either forward or backward!
 - ▶ A 'wait for' command for either your ultrasonic or touch sensor
- ▶ **And for each turn in the maze you need**
 - ▶ A single move command that performs your turn
- ▶ **That's it!!**
- ▶ **One common mistake**
 - ▶ Your very last move command that gets you out of the maze should not be an unlimited command if you do not have a 'wait for' command after it!!
 - ▶ Your robot does not understand the 'unlimited' duration if there is no 'wait for' command after the move command.
 - ▶ So, if you only have to use one more straight command to get out of the maze, just set a number of rotations that is large enough for you to leave the maze
 - ▶ Or, if you really want to use unlimited, then add one more 'wait for' command at the end!



What if I want to use wait for light or wait for sound?

- ▶ These will not be very easy to use to follow a maze, but you may want to use them for other programming projects.

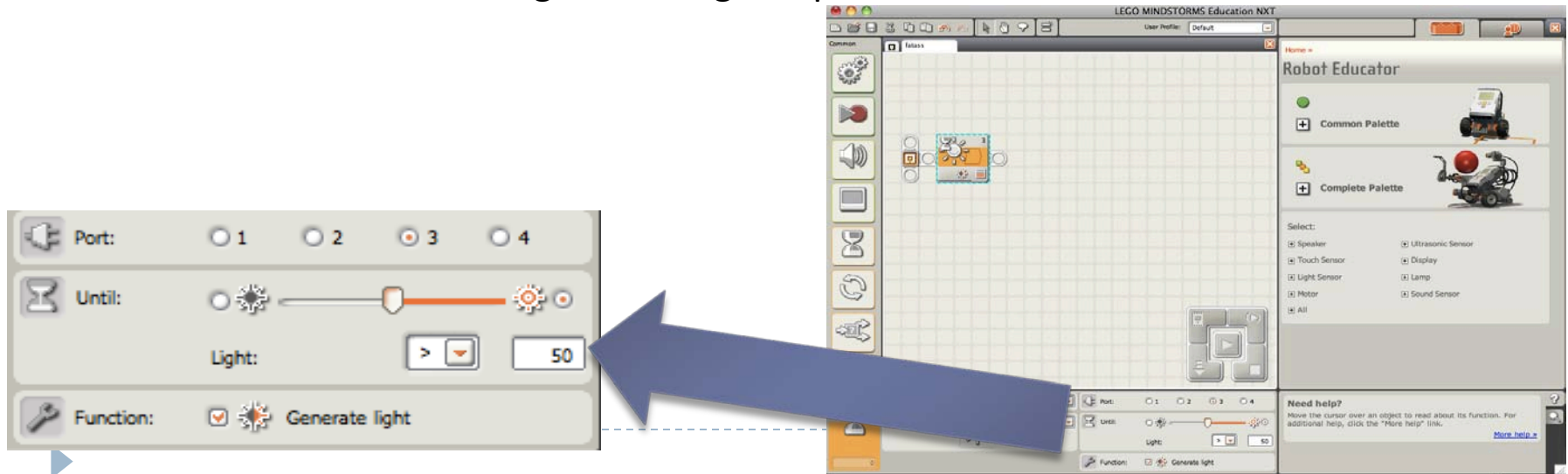


- ▶ Wait for light:
- ▶ You can use your light sensor in two ways:
 - ▶ To detect how much light there is in the surroundings
 - ▶ To detect what percentage of light sent out by the light sensor is reflected back into it.

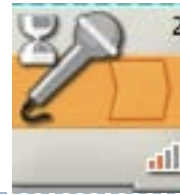


Wait for light

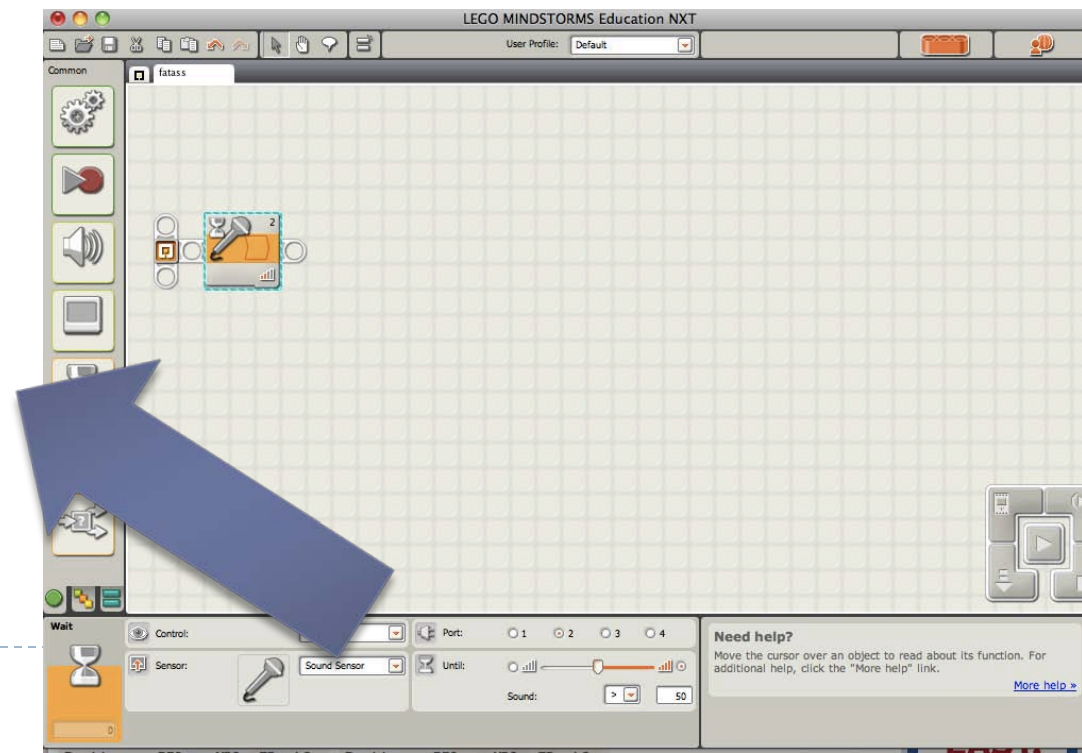
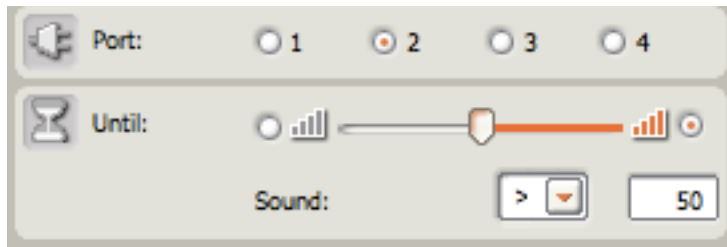
- ▶ Let's examine the options that 'wait for light' presents:
- ▶ You can determine with the slider bar, how sensitive your light sensor is.
- ▶ If you give it a small number, the light sensor will be triggered by lights that may not be really bright. The larger your number, the more bright a light has to be to trigger the sensor.
- ▶ If the 'generate light icon has a check mark, it will see how much of its generated light is reflected back to it.
- ▶ If you want to just check how much light there is in your surroundings, make sure there is no check in the 'generate light' option



Wait for sound



- ▶ The sound sensor is activated based on the volume of the sound it hears.
- ▶ If you slide the bar higher, the sound sensor will need a louder and louder sound to activate. If you slide the bar lower, even quiet sounds will activate the sound sensor



-
- ▶ While these two sensors may not be very useful when solving a maze, they come in handy when trying some other projects that may interest you

