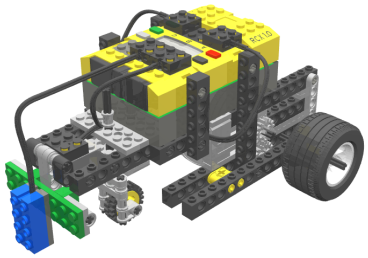
Name: _____

NQC – Light Sensor and More Sounds



Just like the touch sensor, you will need to use the SetSensor command to tell the RCX you are using a light sensor on your robot.

```
SetSensor(SENSOR_2, SENSOR_LIGHT);
```

Put it in the beginning of `task main`.

SetSensor for the light sensor

The light sensor can be used in a similar way to the touch sensor but with one big difference: the value of the sensor is now going to be a number between 0 and 100, usually somewhere between 30 and 60. Start with this simple program:

```
task main()
{
  SetSensor(SENSOR_2, SENSOR_LIGHT);
}
```

Determining a threshold value

1. Download and run the program above.
2. Press the View button on the RCX until there is a small arrow on the display pointing to the sensor 2 port. The display will now show you the current value of the sensor.
3. Using the test track, put your robot so that the light sensor is on the white portion of the track and write down the value.
4. Put your robot so the on something black on the test track and write down that value.
5. Average the two numbers together, this number will be your threshold. Anything greater then the threshold will be considered white, and anything less will be considered black.

Using the light sensor

Just like the touch sensor, you can use the until statement if you want your robot to do something until the light sensor sees a value above or below your threshold. Lets say I find that the best threshold for my robot is 45. What do you think this program will do?

```
task main()
{
  SetSensor(SENSOR_2, SENSOR_LIGHT);
  On(OUT_A+OUT_C);
  until(SENSOR_2 < 45);
  Off(OUT_A+OUT_C);
}
```

Tracking a line

One use of the light sensor is to track lines. The easiest way to do this is to track the edge of the line. Try this for thought:

Program sample	**Description**

`task main()`
`{`
`  SetSensor(SENSOR_2, SENSOR_LIGHT);`
`  On(OUT_A)`
`  until(SENSOR_2 < 45);`
`  Off(OUT_A); On(OUT_C);`
`  until(SENSOR_2 > 45);`
`  Off(OUT_C);`
`}`
	What will this program do assuming that 45 is a good threshold for the robot, if you try this, make sure you determine the best threshold for your robot.

See if you can put this code into a loop so it will track a line.

Playing more sounds

Okay, you have all been clamoring to make more sounds so here are all the sound you can make using PlaySound:

SOUND_CLICK	SOUND_DOUBLE_BEEP	SOUND_DOWN
SOUND_UP	SOUND_LOW_BEEP	SOUND_FAST_UP

Playing music

The RCX can also play notes so you can even make your RCX play a song. To Play a note, you need the PlayTone command. To Play an A note, like on a Piano, you would use the command:

PlayTone(440, 50);

To play a the middle C from the piano, you would do:

PlayTone(523, 50);

The numbers 440 and 523 are the frequencies of the A and C notes on the piano. So to play a song it seems you would need to know all the frequencies of all the notes that your song needs. Fortunately, Bricx Command Center can make this part much easier.

Using Brick Piano

1. Make sure your robot is in front of the IR tower. From the Tools menu, select Brick Piano.
2. Click on the keys of the piano. You should now here the notes played on the RCX.

To Record a Song into a NQC Program

1. On the Brick Piano window, press the Clear button to start a new song.
2. Click on the keys to record and play the notes, you can also alter the length of the notes and even insert rests.
3. Click on the Copy button. This will copy the PlayTone commands for your song onto the clipboard.
4. Now go back to your NQC program and put the insertion point (the flashing vertical line) at the very top of the program. From the Edit menu, select Paste. It will now look something like:

```
#define __NOTETIME 10
#define __WAITTIME 12

  PlayTone(349, 4*__NOTETIME); Wait(4*__WAITTIME);
  PlayTone(330, 4*__NOTETIME); Wait(4*__WAITTIME);
  PlayTone(294, 4*__NOTETIME); Wait(4*__WAITTIME);
```

5. Now put the PlayTone commands inside a task of their own by typing something like this before the first PlayTone command:

```
task PlaySong()
{
```

6. You will also need a clocking } after the last PlayTone command, the top of your program should now look like this

```
#define __NOTETIME 10
#define __WAITTIME 12
task PlaySong()
{
  PlayTone(349, 4*__NOTETIME); Wait(4*__WAITTIME);
  PlayTone(330, 4*__NOTETIME); Wait(4*__WAITTIME);
  PlayTone(294, 4*__NOTETIME); Wait(4*__WAITTIME);
}
```

7. From your task main, add a `start PlaySong;` command.

If you put the `PlayTone` commands inside a `while(true)` loop, you can make the task play the song over and over again.