Unit 3: BRIGHTNESS, IF, and WHILE — Application: Lite Music

In this application, you will write a program to control the sound coming from the speaker based on changing the brightness sensed by the light sensor – hence, 'Lite Music'!

Objectives:

- Write a program that converts brightness into sounds
- Review musical note frequencies and the twelfth root of 2 principle

---

Write a program that reads the BRIGHTNESS of a light sensor and plays a different sound depending on the brightness. There are two possible options for the sound:

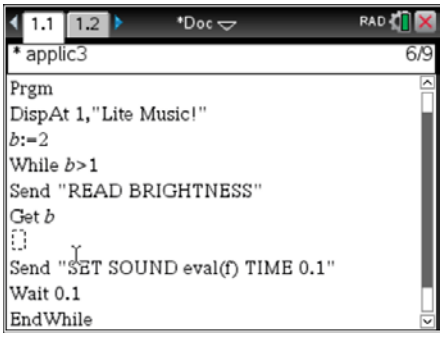
- Play a frequency in the audible range (perhaps 100 Hz – 1000 Hz).
- Play a musical note (one of the specific harmonious sounds found on a piano or other musical instrument).

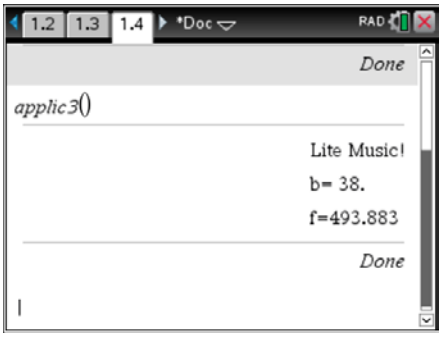
The first option would just play what sounds like 'noise'. The second option will sound more like music, but the mathematics is a bit more complex.

This program makes the TI-Innovator™ Hub behave like a *theremin*. You can vary the amount of light visible to the BRIGHTNESS sensor by adding light (using a flashlight) or removing light (partially covering the sensor).

Getting Started

1. Start a new program, and name it applic3
2. Add **DispAt 1, "Lite Music!"**.
3. Initialize the variable **b** by adding the statement **b:=2**.
4. Add a **While** loop to **READ** the **BRIGHTNESS** sensor and **Get** its value into a variable **b**.
5. Add the statement to play a sound.
6. Notice that we're using the variable ***b*** for reading BRIGHTNESS and the variable ***f*** for playing the SOUND.





Your task is to complete the missing code that converts the BRIGHTNESS into an audible *sound* or a *musical note*.

For sound, use a frequency between 100 and 1000 Hz (or two frequencies of your choice).

For musical notes, try a range starting with A1 (55 Hz) and going up 50 notes. (Refer to the activity studied in Unit 2, Skill Builder 3 (program sound2) which played the 12 notes in an octave.)

For the musical notes, you will need to convert your value to a whole number so that a note 'number' is correctly represented. You can use either the **int( )** function or the **round(,0)** function.

> **x:=int(x)** or **x:=floor(x)** gives the greatest integer less than or equal to **x**.
> **x:=round(x,0)** rounds **x** to the nearest integer.

Teacher Tip:

For just playing a sound, students need to convert B in (0,100) to F in (100,1000).

B	F
0	100
100	1000

Calculate the slope of the line containing these two points, and then write the equation for F in terms of B.

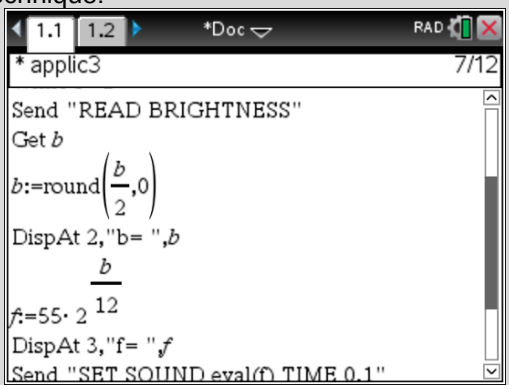
M = (1000-100)/(100-0) ==> 9

so

F = 9*B+100

For playing musical notes, recall the <var>*2^(1/12) property of note intervals. A1=55 Hz, and we want 50 notes. Also, remember that the note number must be an integer, so we must use either int() or round(). We use a 2-step computation to make it clear:

b:=round(b/2,0) --note number

f:=55*2^(b/12) --note frequency

Here's this conversion technique: