Unit 3: Brightness, if and while with the TI-Innovator™ Hub — Application: Lite Music

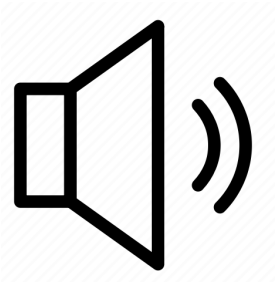
In this application, you will control sounds using the brightness sensor. There are three parts to this project:

1. Light tones (frequencies)
2. Notes using tones (frequencies of notes)
3. Notes using a list of note "names"

Objectives:

- Set the **brightness.range()** so that the value is suitable for making sounds
- Play sounds or musical notes by varying the brightness.

In an earlier lesson, you learned about **sound.tone()** and **sound.note()** using the TI-Innovator Hub. In this lesson, you will use the brightness sensor to create 'noise' and 'music' (sometimes it is hard to tell the difference!).

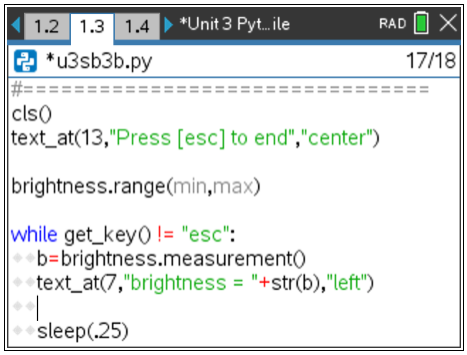


Teacher Tip: Musical notes have specific frequencies of the form 55*2**(k/12) where k is a whole number (k>=0). Tones can be any frequency.

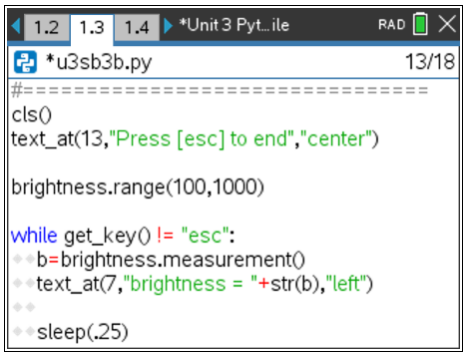
Part 1: Light Tones

1. Again, use the original 'brightness meter' program from the first lesson in this unit. Make another copy of the program using **menu > Actions > Create Copy…**.

   Next, you must decide what **brightness.range()** would be appropriate to use for sounds.



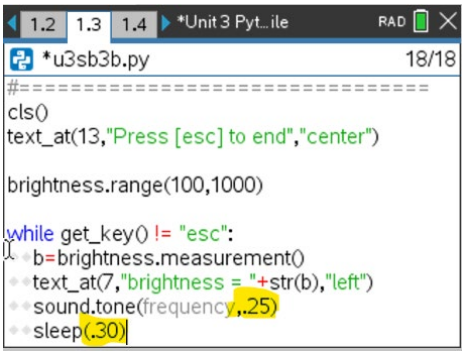
2. For *tones* we can use any frequency between 0 and 8000 Hz, but many of those frequencies are too high or too low for humans to hear. Start with a range of (100,1000) and adjust to your liking.



3. Add the **sound.tone()** statement *below* the **brightness.measurement** statement and use the variable **b** for the **frequency** argument. Set the sound's **time** to your preference and use the same value in the **sleep()** statement so that the TI-Innovator Hub and the handheld are in sync.

   If you like, try making the **sleep()** value a little *larger* than the tone time value. This puts a little silent gap between sounds.

   Test your program now and then adjust the numbers you used.



Teacher Tip: The next section incorporates music theory from Unit 1.

Part 2: Musical Notes Using Frequencies

1. In the five octaves pictured, there are a total of 60 notes (12 per octave). Note A1 (A in the first octave) has frequency 55 Hz.

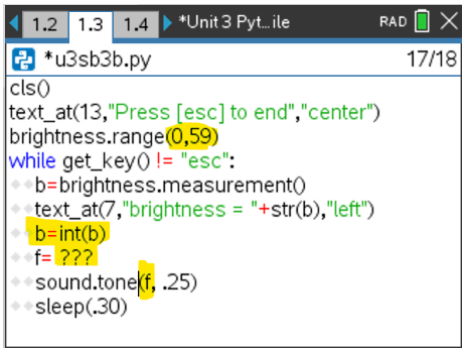
   Subsequent notes have frequency **55 * 2 ** (k/12)**, where k is the note number *after* A1. A1 is note number zero: k=0 → 2**(0/12) = 1.

Notes	Frequency (octaves)				
A	55.00	110.00	220.00	440.00	880.00
A#	58.27	116.54	233.08	466.16	932.32
B	61.74	123.48	246.96	493.92	987.84
C	65.41	130.82	261.64	523.28	1046.56
C#	69.30	138.60	277.20	554.40	1108.80
D	73.42	146.84	293.68	587.36	1174.72
D#	77.78	155.56	311.12	622.24	1244.48
E	82.41	164.82	329.64	659.28	1318.56
F	87.31	174.62	349.24	698.48	1396.96
F#	92.50	185.00	370.00	740.00	1480.00
G	98.00	196.00	392.00	784.00	1568.00
Ab	103.83	207.66	415.32	830.64	1661.28

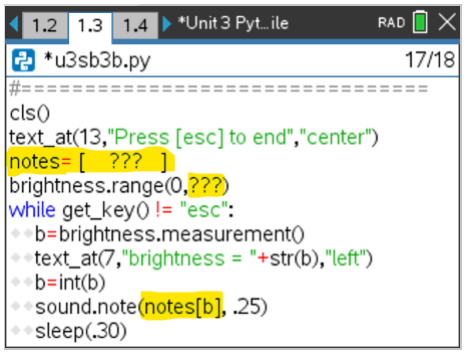
2. To play 'notes,' modify your program.
   Change the **brightness.range()** to be 0…59.
   brightness.measurement() produces a decimal but we only want integers so convert **b** to an integer using **b = int(b)**.
   int() is found on **menu > Built-ins > Type**.

   Calculate a note's frequency using **f = 55*2**(b/12)**.
   Use the variable **f** in the **sound.tone()** statement for *frequency*.

   Try the program again. Some notes might be too high or too low. What can you do to limit the range of the notes?



Part 3: Notes Using a List



1. Recall that the sound object can also use note "names".
   At the top of your program (before the **while** loop), make a list of note "names" as you did in an earlier lesson:

   notes = ["C5", "D5", "E5", …]

   Note names are any of the letters **ABCDEFG** followed by a number from **12345**.
   Set the **brightness.range()** to be (0, *# of notes in your list -1*).
   Convert the variable **b** to an integer.
   Use the variable **b** as the index of the notes list:

   sound.note(notes[b], .25)

   Run the program now.

   This program plays the notes in the list with low brightness using the beginning notes and high brightness using the notes from the end of the list.

> **Teacher Tip:** Here is a partial list of notes. Insert 'F' (flats) and/or 'S' (sharps) where appropriate.
>
> notes = ['A1', 'B1', 'C1', 'D1', 'E1', 'F1', 'G1', 'A2', 'B2', 'C2', 'D2', 'E2', 'F2', 'G2', 'A3', 'B3', 'C3', 'D3', 'E3', 'F3', 'G3', 'A4', 'B4', 'C4', 'D4', 'E4', 'F4', 'G4', 'A5', 'B5', 'C5', 'D5', 'E5', 'F5', 'G5']