

The TI-Innovator Piano

Mini-Project 3: Play the Piano using the Brightness Sensor

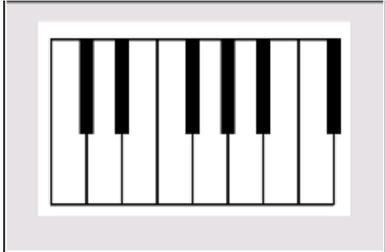
In this third mini-project, you will import the piano image. You will use the brightness sensor on the TI-Innovator Hub to measure the brightness level in the room. Based on the brightness, draw a circle on the correct key and play the corresponding note.

Objectives:

- import PIANO to draw your keyboard
- use getKey to retrieve key press values
- use IF statements to make decisions
- use LISTS to store information
- use a WHILE statement to repeat code

The Piano Project Overview:

After completing a series of 5 mini-projects, you will have a few different ways to play a piano. You will be able to play the piano using the keys on your handheld, using the brightness sensor on the TI-Innovator Hub, using a separate ultrasonic sensor or using the ultrasonic sensor on the TI-Rover. *If you are unfamiliar with using the getKey command, you should do mini-project 1 "Detect which keys are pressed" in the Maze project first.*



Mini-project order:

1. Piano setup
2. Play the piano using the keypress event.
- 3. Play the piano using the brightness sensor on the TI-Innovator™ Hub**
4. Play the piano using an ultrasonic sensor on the TI-Innovator™ Hub
5. Play the piano using the ultrasonic sensor on the TI-Innovator™ Rover



Teacher Tip:

If the TI-Innovator hub doesn't seem to be working, check to make sure the calculator has at least 50% charge and is connected to the calculator.

1. Do you recall how to use the brightness sensor on the TI-Innovator Hub?

What is the smallest value? What is the largest value?

Lets do a little investigation to find out.

Create a new program named: PIANOBRT

Type the following code

```
:0→K
:While K≠45
:getKey →K
:Send("READ BRIGHTNESS ")
:Get(B)
:Output(6,1,B)
:End
```





Hint: A light source such as a cell phone flashlight or lamp might be helpful. Remember to shine it on the brightness sensor on the hub.

- The piano is a background image. Therefore, we only need to draw it once. You will insert the base code from the piano project above the code you just wrote.

Arrow up to your first line of code. Press  

Then choose Insert Line Above (3 times)

```

:
:
:
:0→K
:While K≠45
:getKey →K
:Send("Read Brightness ")
:Get(B)
:Output(6,1,B)
:End

```

- On the **first line**, use recall to import the keyboard code

```

rcl (   )
prgm
EXEC
Select PIANO
Press the enter key

```

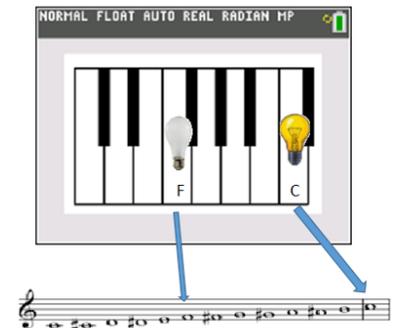
Execute your code. It should draw the keyboard first. Then display the brightness sensor data. The display will remove your keyboard, which is ok for now. Make sure you don't have any errors.

- You will now edit the brightness sensor code inside the loop. You will code two frequencies, one for low light and one for more light. We will use 10 in the demo code, but you can use number you like based on your trial. If there is a lot of light, we will play middle C with a frequency of 261 Hz. If there is low light, we will play a lower note F at 174Hz

```

:
:
:
:0→K
:While K≠45
:getKey →K
:Send("READ BRIGHTNESS ")
:Get(B)
:Output(6,1,B)
:End

```





- In the While loop used to get the brightness, delete the last two lines of the loop

```

:0→K
:While K≠45
:getKey →K
:Send("READ BRIGHTNESS ")
:Get(B)

```

You will use an IF to play a low frequency for low light and a high frequency for more light. We will use 10 in the demo code, but you can use any number you like based on your trial from step 1. If there is a lot of light, we will play middle C with a frequency of 261 Hz. If there is low light, we will play a lower note F at 174Hz.

Code the IF statement to play a frequency of 261Hz if the brightness is below 10 otherwise play a frequency of 174 Hz.

- Does your loop look similar to the one below?

```

:0→K
:While K≠45
:getKey →K
:Send("READ BRIGHTNESS ")
:Get(B)
:If B<10
:Then
:Send("SET SOUND 174")
:Else
:Send("SET SOUND 261")
:End
:End

```

Did you try your code? Did you fix any errors? It is very important you check your code as you go. It makes finding and fixing errors easier.

- Our piano keyboard has 14 keys on it. You could make 14 if statements to play each note based on the light level. For example, you could say

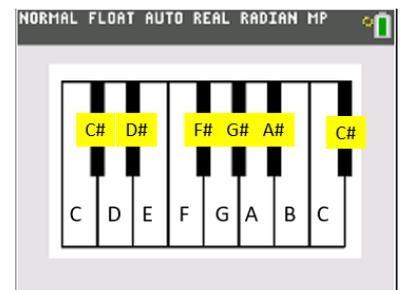
```

:If B>5 and B <15
:Then
:Send("SET SOUND 261")
:End

```

That would take you 14*5 = 70 lines of code!!

How could we use a mathematical relationship to help decrease the number of lines of code?





8. In the first piano project we used the equation $f = 261.64 * 2^{n/12}$ where f is the frequency and n was a whole number exponent?

Depending on your light source, the brightness (B) can range from 0 to 100. Therefore, we need to split the range of values over 14 keys. If we split the values evenly, we get the equation

$$\text{number of keys} * x = \text{brightness range}$$

$$14x = 100$$

$$x = \frac{100}{14} = \frac{50}{7} \approx 7.14$$

That means the frequency should change each time the brightness increases by 7.14.

C should play when the brightness is from 0 to 7.14.

C# should play when the brightness if from 7.15 to 14.28.

D should play when the brightness is from 14.29 to 21.42...

9. $F = 261.64 * 2^{N/12}$

Let's say B, the brightness, is 18.

$$18/7.14 \approx 2.52$$

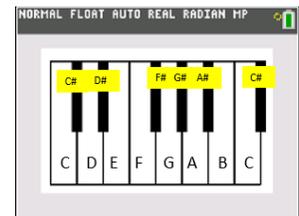
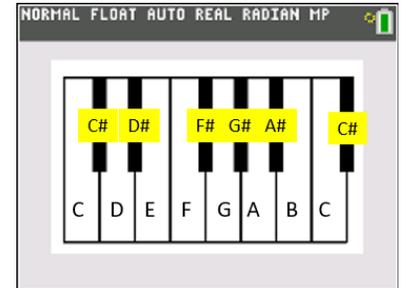
That should play D which had an exponent of 2.

Let's say B, the brightness, is 70.

$$70/7.14 \approx 9.80$$

That should play A which had an exponent of 9.

To get N for the formula, you need the integer part of result. To do this you will use the command `iPart()` located in the Math menu under Num. It returns just the integer part of a result. How do you think you should modify your code to play the right note?



Note	Exponent	Range	
C	0	0	7.14
C#	1	7.15	14.28
D	2	14.29	21.42
D#	3	21.43	28.56
E	4	28.57	35.7
F	5	35.71	42.84
F#	6	42.85	49.98
G	7	49.99	57.12
G#	8	57.13	64.26
A	9	64.27	71.4
A#	10	71.41	78.54
B	11	78.55	85.68
C	12	85.69	92.82
C#	13	92.83	100



10. Does your loop look similar to the one below?

```

:0→K
:While K≠45
:getKey →K
:Send("READ BRIGHTNESS ")
:Get(B)
:iPart(B/7.14)→N
:Send("SET SOUND eval(261.14*2^(N/12))")
:End

```

Did you try your code? Does it work?

*If you have a darker room, you might have to change your value of 7.14 to a different number to get the higher notes.

11. Insert (Alpha Graph) 2 blank lines above the last WHILE statement

In order to draw a point on the selected key, you need to know if the exponent numerator, N, matches a sharp or a flat.

Store these values in 2 different lists.

Natural List:

```
{0, 2, 4, 5, 7, 9, 11, 12} →L1
```

Sharp List:

```
{1, 3, -1, 6, 8, 10, -1, 13} →L2
```

Notice there are two negative ones in the list. We need to save a space for the missing keys. In the code, if the value in the list is -1 we won't draw it.

12. Recall in the piano mini-project 1 you wrote the code to highlight all the nature keys.

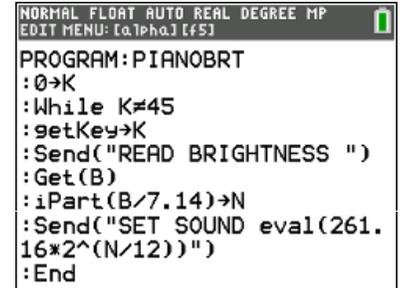
```

For(A,0,7)
Pt-On(25+30*A, 30, 2, MAGENTA)
End

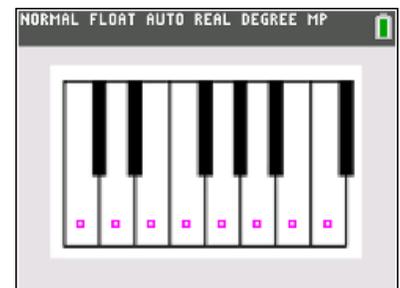
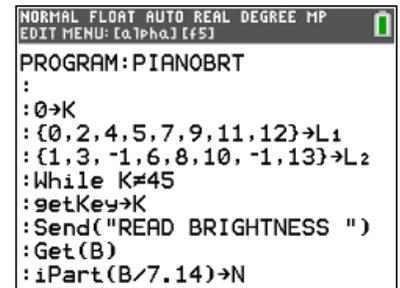
```

Natural List:

```
{0, 2, 4, 5, 7, 9, 11, 12} →L1
```



Note	Exponent	Range	
C	0	0	7.14
C#	1	7.15	14.28
D	2	14.29	21.42
D#	3	21.43	28.56
E	4	28.57	35.7
F	5	35.71	42.84
F#	6	42.85	49.98
G	7	49.99	57.12
G#	8	57.13	64.26
A	9	64.27	71.4
A#	10	71.41	78.54
B	11	78.55	85.68
C	12	85.69	92.82
C#	13	92.83	100





You will combine these two concepts to highlight the appropriate key.

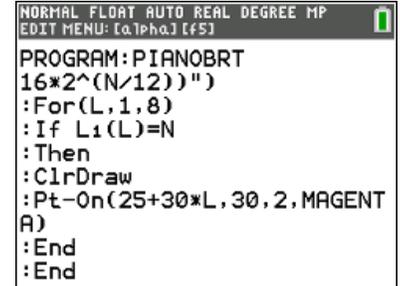
$$L1(1) = 0, L1(2) = 1, L1(3) = 4, L1(4) = 5 \dots L1(8) = 12$$

Insert (Alpha Graph) 7 blank lines above the last End statement in your code.

```

:For(L, 1, 8)
:If L1(L) = N
:Then
:ClrDraw
:Pt-On(25+30*L, 30, 2, MAGENTA)
:End
:End

```



13. Recall in mini-project 1 you wrote the code to highlight all the sharp keys.

```

:For(A,0,7)
:If A≠2 and A ≠6
:Pt-On(41+30*A, 90, 2, MAGENTA)
:End
:End

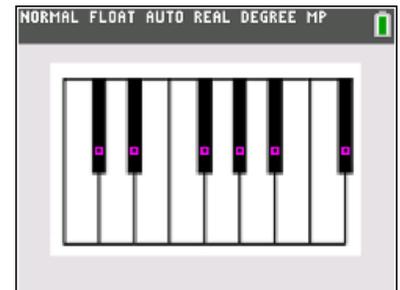
```

Sharp List:

```

{{1, 3, -1, 6, 8, 10, -1, 13} →L2

```



You will combine these two concepts to highlight the appropriate key.

$$L2(1) = 1, L2(2) = 3, L2(3) = -1, L2(4) = 6 \dots L2(8) = 13$$

Insert (Alpha Graph) 7 blank lines above the last End statement in your code.

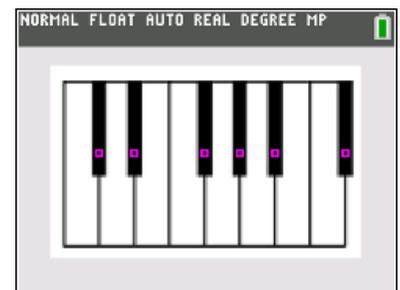
Can you write the FOR loop and IF statement to color the appropriate sharp key? Check your answer with the next step.

14. Is your code similar to the code below?

```

:For(L, 1, 8)
:If L1(L) = N and L1(L) ≠ 0
:Then
:ClrDraw
:Circle(39+30*L, 90, 10, MAGENTA)
:End
:End

```



Can you play a song with your piano?



```
NORMAL FLOAT AUTO REAL DEGREE MP
EDIT MENU: [alpha][f5]
PROGRAM:PIANOBRT
:For(L,1,8)
:If L2(L)=N and L2(L)≠-1
:Then
:ClrDraw
:Pt-On(41+30*L,90,2,MAGENT
A)
:End
:End
:End
```

Teacher Tip:

Demo Code

```
NORMAL FLOAT AUTO REAL DEGREE MP
EDIT MENU: [alpha][f5]
PROGRAM:PIANOBRT
:0→Xmin
:264→Xmax
:0→Ymin
:164→Ymax
:FnOn
:PlotsOff
:BackgroundOn Image1
:ClrDraw
:DispGraph

:
:0→K
:{0,2,4,5,7,9,11,12}→L1
:{1,3,-1,6,8,10,-1,13}→L2
:While K≠45
:getKey→K
:Send("READ BRIGHTNESS ")
:Get(B)
:iPart(B/7.14)→N
:Send("SET SOUND eval(261.
16*2^(N/12))")
:For(L,1,8)
:If L1(L)=N
:Then
:ClrDraw
:Pt-On(25+30*L,30,2,MAGENT
A)
:End
:End
:For(L,1,8)
:If L2(L)=N and L2(L)≠-1
:Then
:ClrDraw
:Pt-On(41+30*L,90,2,MAGENT
A)
:End
:End
:End
```