The TI-Innovator™ Piano

Mini-Project 4: Play the Piano Using the Ultrasonic Sensor

In this fourth mini-project, you will import the piano code from the brightness sensor. You will use the ultrasonic sensor and the TI-Innovator Hub to measure distances to the sensor. Based on the distance, draw a circle on the correct key and play the corresponding note.

Objectives:

- import PIANOBRT to draw your keyboard and import the loop
- use the ultra-sonic range sensor and the TI-Innovator Hub
- 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.

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. **<u>Play the piano using an ultrasonic sensor and the TI-Innovator Hub</u>**
5. Play the piano using the ultrasonic sensor on the TI-Innovator Rover

Tech 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. If you don't have a separate ultra-sonic sensor but you have the TI-Innovator Rover, you can skip this mini-project and go straight to mini-project 5.

1. Connect your ultrasonic ranger to IN 1 on your TI-Innovator.

   Make sure it is plugged into IN 1 in the TI-Innovator Hub.

2. Let's make sure your ranger is working. *Create a temporary project named T.*

   In order to get information from the ultrasonic ranger, you must first connect the TI-Innovator to the ranger. We didn't have to do this with the brightness sensor because it is built into the hub.

   Type the following lines of code.

   You will get most of the code in the HUB menu. ***Do not type the code, get the code from the menu***

   :Send("CONNECT RANGER 1 to IN 1")

   **Send("Connect* *Press PRGM then it is listed under HUB*
   **to* *is located in the HUB menu under Settings*
   **IN 1* *is located in the HUB menu under Ports*

   Then type the rest of the program:

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

   Execute your code. Hover your hand over the sensor. What types of values do you get?

3. On the first line, use recall to import the keyboard code
   rcl (2nd sto→)
   prgm
   EXEC
   Select PIANOBRT
   Press the enter key

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

4. Remove the three lines of code that uses the brightness sensor. You will insert code to read from the ultrasonic ranger.

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

```
:
:0→K
:While K≠45
:getKey→K
:
:
:
:Send("SET SOUND eval(261*
2^(N/12)))
```

5. Insert the statements to read from the ultrasonic ranger.

```
:0→K
1 :Send("CONNECT RANGER 1 to IN 1")
:While K≠45
:getKey →K
2 :Send("READ RANGER 1")
3 :Get(D)
:
:Send("SET SOUND eval(261.14*2^(N/12))")
:End
```

```
NORMAL FLOAT AUTO REAL DEGREE MP
EDIT MENU: [alpha] [f5]
PROGRAM:PIANODST
 1 ")
:While K≠45
:getKey→K
:Send("READ RANGER 1")
:Get(D)
:
:Send("SET SOUND eval(261.
16*2^(N/12))")
:For(L,1,8)
```

Using the numbers from your investigation and mini-project 3, what do you think goes in the empty line to produce the various notes?

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

```
:0→K
:Send("CONNECT RANGER 1 to IN 1")
:While K≠45
:getKey →K
:Send("READ RANGER 1")
:Get(D)
:iPart(D*100) →N
:Send("SET SOUND eval(261.14*2^(N/12))")
: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.