The TI-Innovator Piano

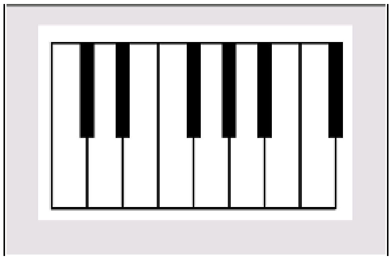
Mini Project 1: Piano Setup

In this first piano mini-project, you will import the piano image. You will learn how to draw a circle to represent a piano key press. Once you have this module coded, it will serve as base code for all the other piano projects.

Objectives:

- import a piano picture
- use if statements to make a selection
- draw a circle to indicate which piano key is pressed
- use a For 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 simulation on your calculator. 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 might do Mini-Project 1 “Detect which keys are pressed” in the Maze project before you start the Piano project.

Activity 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 an the TI-Innovator™ Hub
5. Play the piano using the ultrasonic sensor on the TI-Innovator™ Rover

Teacher Tip:
If students are unfamiliar with the numeric values of the keys on the keyboard and how to use their values, Activity 1 “Detect which keys are pressed” in the Maze project might be helpful before Activity 2. It teaches students how to use a While statement to continuously check for keypress values. It also lets students explore and create a map of key numbers for the keyboard. If students don’t have this knowledge, the code will be provided in the piano project steps to overcome the skill deficit.

1. Connect your calculator to your computer and run TI-Connect CE.

   Open the calculator workspace.

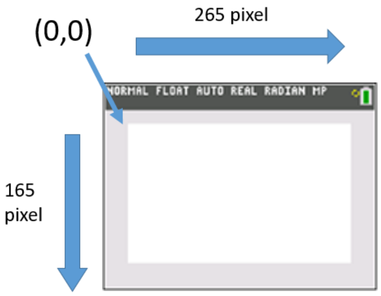
   Import the piano drawing as IMAGE1

   Create a new program named PIANO

2. Your calculator has a width of 165 pixels and a height of 265 pixels.

   By default, the upper left corner is (0, 0).

   This may seem odd, but often in programming as you go down a screen the values INCREASE.

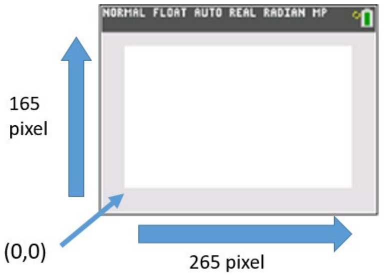


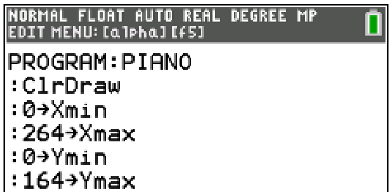
3. You can change the orientation of the coordinate grid so (0, 0) is in the lower left and the value increase as you go up the screen.

   Put ClrDraw as your first line of code (2nd prgm)
   This will clear any old graphics off the screen.

   Set the x values using the code
   :0→Xmin
   :264 →Xmax

   Set the y values using the code
   :0 →Ymin
   :164 →Ymin



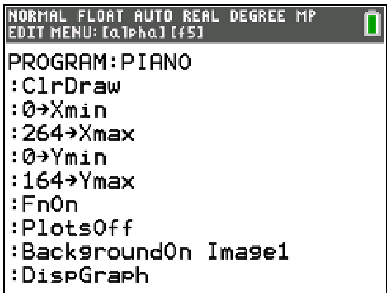


```
PROGRAM:PIANO
:ClrDraw
:0→Xmin
:264→Xmax
:0→Ymin
:164→Ymax
```

****
Xmin, Xmax, Ymin, Ymax are located in Vars – Windows

4. You need to:
   turn functions off
   turn plots off
   set the background to your image

   The following code will accomplish the three tasks:
   :FnOff
   :PlotsOff
   :BackgroundOn Image1
   :DispGraph

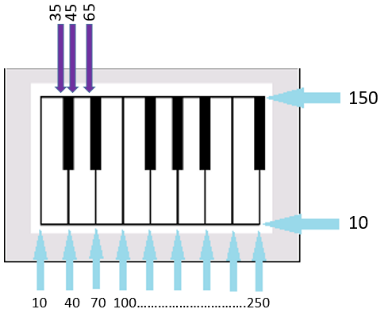


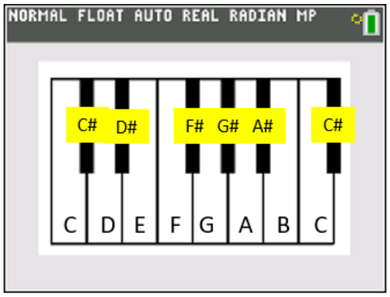
```
PROGRAM:PIANO
:ClrDraw
:0→Xmin
:264→Xmax
:0→Ymin
:164→Ymax
:FnOn
:PlotsOff
:BackgroundOn Image1
:DispGraph
```

****
BackgroundOn is located in Draw – Background
Images are in Vars- Picture & Background
FnOff is located in Vars – Y-Vars – On/Off
PlotsOff is located in statPlot
DispGraph is located in PRGM – I/O

5. The natural (white) keys start at x = 10. Each natural key has a width of 30 pixels. You have eight natural keys in your picture.

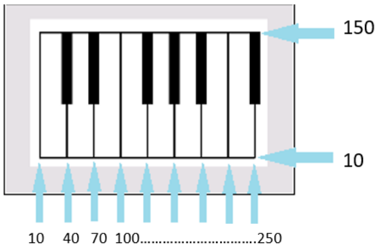
   The sharp (black) keys start at x = 35. Each sharp key has a width of 10 pixels. They are also 30 pixels apart. There are six sharp keys in your picture.



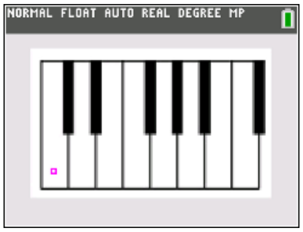


6. To represent a pressed key, you'll draw a point using the point on command.

   To draw the point, the code is Pt-On(X, Y, Style, Color) where the point (X,Y) is the center of the point. We will use Style = 2, but feel free to explore style values from 1-7.

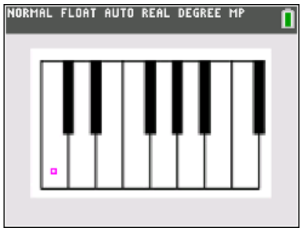


What would the line of code look like to draw the point to the right?

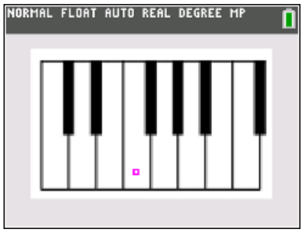


7. Did your code look similar to the code below?

   : Pt-On(25, 30, 2, MAGENTA)



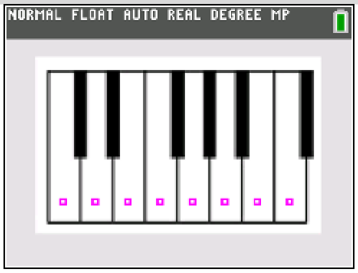
What would the line of code look like to draw the square to the right?



Teacher Tip:
:Circle(25 + 30*3, 30, 10, MAGENTA)

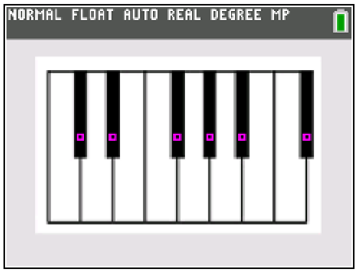
8. If you want to draw squares on all the natural keys, you could write eight different lines of code, or you could write a For loop similar to the one below.

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



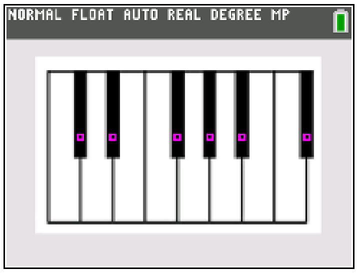
Write a For loop to draw squares on all the sharp keys.
Hint: you'll need an IF statement inside your loop to skip the two missing keys

.



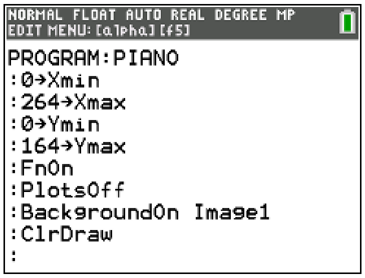
9. Did your code look similar to the code below?

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



Now you know how to draw points on all the keys to represent playing the piano. You will utilize this skill in the next activity which has you incorporate keypress values from the keypad to draw different points. You will then use keypress values to play the correct note for the piano key.

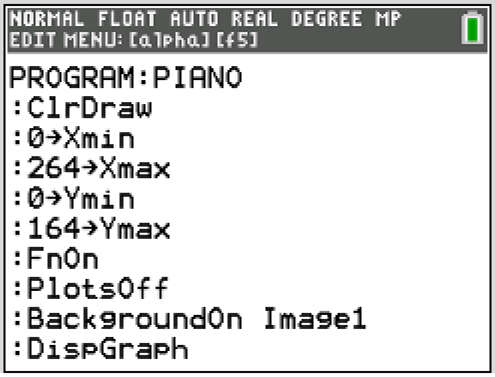
Delete the loops you have written but remember the formula format. You will use the formula later. The only base code you'll need for the next project is shown on the right



```
PROGRAM:PIANO
:0→Xmin
:264→Xmax
:0→Ymin
:164→Ymax
:FnOn
:PlotsOff
:BackgroundOn Image1
:ClrDraw
:
```

Teacher Tip:
Students will import this code into the next four projects. Students may retype the 8 lines of code instead of importing the code.

In the next project, students will add to the drawing skills learned in this activity. They will use the getKey function, Ifs and lists to help draw the appropriate circle and play the correct note.



```
PROGRAM:PIANO
:ClrDraw
:0→Xmin
:264→Xmax
:0→Ymin
:164→Ymax
:FnOn
:PlotsOff
:BackgroundOn Image1
:DispGraph
```