



Name: _____

Assessment Air Guitar

- 1) What does capacitance measure?
 - a) The speed of light
 - b) The ability to hold an electric charge
 - c) The frequency of sound waves
 - d) The temperature of an object

- 2) Which component senses the presence of your hand in the air guitar circuit?
 - a) Jumper wires
 - b) Light sensor
 - c) Foil squares
 - d) Temperature sensor

- 3) What is the purpose of the 330 Ohm Resistors in the circuit?
 - a) To connect the Pico to the foil squares
 - b) To prevent the LED bars from burning out
 - c) To increase the voltage of the LED bars
 - d) To connect the alligator clips to the breadboard

- 4) Why do we use a really big 1 MOhm resistor in the circuit?
 - a) To allow the foil to charge faster
 - b) To slow down the charge from draining back to the ground pin
 - c) To connect the LED bars to the Pico
 - d) To increase the sensitivity of the Pico

- 5) What happens to the graph when you touch the back of the foil square connected to GPO?
 - a) The blue line jumps higher
 - b) The teal line jumps higher
 - c) The graph stops updating
 - d) The graph displays an error message

- 6) Explain how you could use the air guitar you built in a real-world application.



Name: _____

7) Describe an improvement or additional feature you would add to the air guitar project and explain how it would enhance the experience.

8) Look at the Python code below that you generated using Blockly. Can you change the code so that instead of a piano sound, it plays a guitar sound when you touch the foil square? Explain what changes you made and why.



Name: _____

Answer Key Air Guitar

- 1) B - The ability to hold an electric charge
- 2) C - Foil squares
- 3) B - To prevent the LED bars from burning out
- 4) B - To slow down the charge from draining back to the ground pin
- 5) A - The blue line jumps higher
- 6) *Example:* The air guitar uses capacitive sensing to detect the presence and movement of a hand. I could use this technology in real-world applications such as touchless control interfaces for medical devices, where it is essential to avoid physical contact, or in smart home systems for gesture-based control of lights and appliances. Additionally, it can be used in interactive installations or performances where users can create music or visual effects by moving their hands in the air.
- 7) *Example:* One improvement could be to add a Bluetooth module to the air guitar project, allowing it to connect wirelessly to a speaker or a computer. This would make the setup more flexible and mobile, enabling users to perform without being tethered to a specific location. Additionally, incorporating different sensor types, such as proximity sensors or accelerometers, could allow for more complex gestures and a wider range of musical expressions. This enhancement would make the air guitar more interactive and engaging, providing a richer user experience.
- 8) *Example:* To change the code so that it plays a guitar sound instead of a piano sound, I modified the `playSound` function call. Specifically, I changed the string from `"instrument|piano|"` to `"instrument|synth|"`. This tells the program to use the synth sound for the notes.

```
## ---- Imports ---- ##
import board
from piper_blockly import *
import time

## ---- Definitions ---- ##
foil_square = None
GP4 = piperCapSensePin(board.GP4, "GP4")

try:
    set_digital_view(True)
```



Name: _____

```
except:
    pass

GP15 = piperPin(board.GP15, "GP15")
GP14 = piperPin(board.GP14, "GP14")
GP13 = piperPin(board.GP13, "GP13")
GP12 = piperPin(board.GP12, "GP12")
GP11 = piperPin(board.GP11, "GP11")
GP10 = piperPin(board.GP10, "GP10")
GP9 = piperPin(board.GP9, "GP9")
GP8 = piperPin(board.GP8, "GP8")
GP7 = piperPin(board.GP7, "GP7")
GP6 = piperPin(board.GP6, "GP6")

def piperBarGraphLED(single, show_value):
    show_value = int(show_value)
    bar_graph_pins = [GP15, GP14, GP13, GP12, GP11, GP10, GP9, GP8, GP7, GP6]
    for pin in bar_graph_pins:
        pin.setPin(0)
    if (show_value <= 0): return
    if (show_value > 10): show_value = 10
    if (single):
        bar_graph_pins[int(show_value) - 1].setPin(1)
    else:
        for nn in range(int(show_value)):
            bar_graph_pins[nn].setPin(1)

GP0 = piperCapSensePin(board.GP0, "GP0")

## ---- Code ---- ##
foil_square = GP4.readCapSenseValue()
while True:
    while (GP0.readCapSenseValue()) < 2000:
        foil_square = foil_square * 3
        foil_square = foil_square + (GP4.readCapSenseValue())
        foil_square = foil_square / 4
        piperBarGraphLED(True, (mapValue(foil_square, 1500, 4000, 0, 10)))
    playSound("instrument|synth|" + str(0.125) + "|" +
str((mapValue(foil_square, 1500, 4000, 220, 880))))

    time.sleep(0.1)
```

Explanation:
To change the sound from piano to guitar, I modified the `playSound` function in the code from:

```
playSound("instrument|piano|" + str(0.125) + "|" +
str((mapValue(foil_square, 1500, 4000, 220, 880))))
```



Name: _____

to:

```
playSound("instrument|synth|" + str(0.125) + "|" +  
str((mapValue(foil_square, 1500, 4000, 220, 880))))
```

This alteration instructs the program to use the synth instead of the piano sound for the notes played when you touch the foil square.