Name: ____________________

Assessment

Rover Sensor Steering

1) What does the Ultrasonic Range Finder do?
   a) It uses light waves to measure distance.
   b) It uses sound waves to measure distance.
   c) It uses radio waves to measure distance.
   d) It uses infrared waves to measure distance.

2) What is the purpose of the GND pin in the wiring diagram?
   a) To provide power to the sensor.
   b) To receive signals from the sensor.
   c) To ground the sensor.
   d) To transmit data to the sensor.

3) In the tutorial, what does setting motor_0 to -30% speed and motor_1 to 30% speed do to the Rover?
   a) It makes the Rover move forward.
   b) It makes the Rover spin in place.
   c) It makes the Rover turn left.
   d) It makes the Rover turn right.

4) At a distance of 15 cm from the sensor, what should the speeds of motor_0 and motor_1 be to make the Rover move straight forward?
   a) -30% for motor_0 and 30% for motor_1
   b) -15% for motor_0 and 15% for motor_1
   c) -60% for motor_0 and 60% for motor_1
   d) 0% for both motors

5) Why is it better for the Rover to have wider turns instead of tight turns when being chased by Zomars?
   a) Wider turns are faster and help maintain speed.
   b) Tight turns use more battery power.
   c) Tight turns can damage the Rover's wheels.
   d) Wider turns are easier to program.

6) Explain how the ultrasonic range finder in the tutorial is similar to the echolocation used by bats.

PIPER

Name: ____________________

7) How could you modify the code so that the Rover's speed decreases as it gets closer to an object, similar to a brake pedal?

8) Look at the Python code below. This code makes the Rover change its speed based on its distance from an object. Can you think of a way to make the Rover stop completely if the object is closer than 5 cm? Modify the code to add this feature. Explain what changes you made and why.

Name: ____________________

Answer Key
Rover Sensor Steering

1) B - It uses sound waves to measure distance.

2) C - To ground the sensor.

3) A - It makes the Rover move forward.

4) B - -15% for motor_0 and 15% for motor_1

5) A - Wider turns are faster and help maintain speed.

6) *Example:* The ultrasonic range finder and echolocation used by bats both work by emitting sound waves that bounce off objects and return to the source. The time it takes for the sound waves to return helps determine the distance to the object. This allows the sensor, like a bat, to "see" using sound instead of light.

7) *Example:* To modify the code so that the Rover's speed decreases as it gets closer to an object, you could map the distance values from the ultrasonic range finder to a speed range. For example, you could set a variable to read the distance from the sensor and then use the **map** block to convert that distance to a speed value. When the distance is small (meaning the object is close), the speed could be low. Conversely, when the distance is large (meaning the object is far), the speed could be high. You would adjust the motor speed variables in the code accordingly to reflect these speed changes.

8) *Example:*

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

## ---- Definitions ---- ##
distance_reading = None
motor_0 = None
motor_1 = None
GP22 = piperDistanceSensorPin(board.GP22, "GP22")

try:
  set_digital_view(True)
except:
  pass
```

Name: ____________________

```
def distance_range(_value):
  if (_value == None):
    return 520
  else:
    return _value

GP0 = piperServoPin(board.GP0, "GP0")
GP1 = piperServoPin(board.GP1, "GP1")


## ---- Code ---- ##
while True:
  distance_reading = distance_range(GP22.readDistanceSensor())
  if distance_reading < 5:
    motor_0 = 0
    motor_1 = 0
  else:
    motor_0 = mapValue(distance_reading, 0, 30, (-30), 60)
    motor_1 = mapValue(distance_reading, 0, 30, (-60), 30)
  GP0.setServoFraction(min(max(motor_0 * 0.0018 + 0.5,0),1))
  GP1.setServoFraction(min(max(motor_1 * 0.0018 + 0.5,0),1))

  time.sleep(0.2)
```

Explanation:

I added an `if` statement to check if the `distance_reading` is less than 5 cm. If it is, I set `motor_0` and `motor_1` to 0, which makes the Rover stop completely. If the distance is greater than or equal to 5 cm, the Rover will behave as before, adjusting its speed based on the distance. This change ensures that the Rover stops if it gets too close to an object, which could prevent collisions.