Distance Gun

Instructor Guide

Level 4

1. **Explanation Tracking Button Clicks Assignment**

   Instructor guide: This Level is all about utilizing the button. Depending on the number of clicks from the user it will perform a different function. The distance gun can perform a different function for every click.The number of clicks starts at 0. Each time the user presses the button it increases. When it hits 4, the number of Clicks resets to 0 and the cycle starts over again.

   Explain the following:

   - **0 Clicks or every 4th Click**
     - At 0 Clicks the screen displays the builder's name as they did in Level 2.
   - **1 Click**
     - At the first click the distance finding points code will run. A separate function is created to calculate the reading.
       - This function returns different values for different instances.
         - If the function cannot find an object in immediate distance it returns 0 or -1. Otherwise it returns the distance it finds.
         - It returns -1 in instances where the average is 0/0, and C integer division returns -1.
       - If 0 or - 1 is returned ⇒ "Object Not Found"
       - If any other value is returned ⇒ Display that value (Average Distance)
   - **2 Clicks**
     - At the second click we find 2 different values (not 0) by taking readings after a delay.
     - By comparing them, we can find if they are greater than, less than, or equal to each other.
       - Using this information we will display 1 of 3 messages ⟶
       - "Object Incoming", Object Outgoing", "Object Stationary"
   - **3 Clicks**
     - At the third click we will take a look at two different readings that are measured after a delay. By finding the difference in the readings and dividing by the delay, the speed of the object in front can be found.
       - The delay is 500 milliseconds.
     - A message is displayed ⟶ "Speed: ---- "

2. Explanation Functions

Instructor guide: To complete this level, knowledge on functions is needed. A function is a statement that performs a specific task. It is a piece of reusable code that can be used anywhere in the program. This is especially advantageous when you need to write the same piece of code multiple times. Functions can perform a certain procedure or return a value. This value can be of any datatype and must be referenced next to the function name.

```
data_type Function_Name() {

    Code

}
```

We will be using a function to calculate the average reading and return the reading detected.

Explain the following:

- `data_type`
  - If the function returns a variable, the type of that variable is the `data_type`.
- `Function_Name()`
  - By writing `Function_Name()` we can perform all the code in the function anywhere in the program. THis makes code a lot easier to write

-
  ```
  () {
      Code
  }
  ```
  - The code inside the brackets {} is the code that will be performed every time the function name is used.
  - This code is like the code you have written in the previous levels

Complete Challenge:

Student Hint: Take a look at each condition individually, and test if they work one after another.

Instructor guide: This challenge will teach the builder to apply what they've learned so far to an overarching project. By performing different actions at each click they will be able to use the Ultrasonic sensor and LCd to their full extent. The challenge is explained in step 1.

Notes:

CODE The following code is a sample solution for Level 4.

```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <UltraDistSensor.h>

UltraDistSensor scanner;
LiquidCrystal_I2C lcd(ADDRESS, 20, 4);

int buttonPin = 2;
int val = 0;
int numClicks = 0;
float distance;

void setup() {

    lcd.init();
    lcd.backlight();
    pinMode(buttonpin, INPUT);
    scanner.attach(9, 10);

}
void loop() {

    lcd.setCursor(0, 0);
    val = digitalRead(buttonPin);

    if(val == HIGH){
        numClicks++;
        if(numClicks >= 4){
            numClicks = 0;
        }
    }
    if(numClicks == 0){

        lcd.print(“FirstName”);
        lcd.setCursor(0, 1);
        lcd.print(“LastName”);

    } else if(numClicks == 1) {

        lcd.clear();
        lcd.print(“Distance: ”);

        int distance = findDistance();
        if(distance == 0 || distance == -1){
           lcd.setCursor(0, 1);
           lcd.print(“Object Not Found”);
        } else {
           lcd.print(distance);
        }

    } else if(numClicks == 2) {

        int firstVal = findDistance();
        delay(200);
        int secondVal = findDistance();
        lcd.clear();

        lcd.setCursor(0, 0);
        lcd.print("Object");

        if(secondVal > firstVal){
           lcd.setCursor(0, 1);
           lcd.print("Outgoing");
        } else if(secondVal < firstVal){
           lcd.setCursor(0, 1);
           lcd.print("Incoming");
        } else if(secondVal == firstVal){
           lcd.setCursor(0, 1);
           lcd.print("Stationary");
        }

    } else if(numClicks == 3){
        int firstVal = findDistance();
        delay(500);
        int secondVal = findDistance();
        lcd.clear();

        int speedC = (firstVal - secondVal)*2;

        lcd.setCursor(0, 0);
        lcd.print("Speed: ");
        lcd.setCursor(0, 1);
        lcd.print(speedC);
        lcd.print(" inch/sec");

   }

   delay(400);

}

int findDistance(){

    float distances = 0;
    int numTimes = 0;

    for(int i = 0; i < 10; i++){
        distance = scanner.distanceInInch();
        if(distance != 0.0){
            distances += distance;
            numTimes++;
        }
    }

    int averageDistance = distances/numTimes;
    return averageDistance;

}
```

Notes:

POSSIBLE PROBLEMS AND ANSWERS

Problem: Function throws an error
Possible Solutions:

1. The value that is returned is not of thesame datatype that was specified
2. The code inside the function is incorrect

Problem: Button is not detected presses accurately
Possible Solutions:

1. Change the delay at the end (Adjust slightly)
2. Hold the press for half a second to make sure it registers (holding for too long will increase the button clicks by more than one)
3. Power pins are not connected to respective (+ or -) pins properly.