

More Sensor Values

Introduction

Last time we looked at the Front Laser Distance and how to use it. This time we'll look into where that value comes from, and the sorts of things we usually deal with when programming robots.

Step 1: More Sensor Data

✓ Activity Checklist

- ☐ The Front Laser Distance block actually comes from a set of many laser values.
- ☐ There's two other blocks we can look at for more information on the laser sensors.

Laser Data at Position 10

Laser Array Length

- ☐ For your understanding; an Array is basically a list. So in this case there is an array of numbers that make up the entire readings of the laser sensors of the robot.
- ☐ Laser Array Length is exactly what you think, it just returns how many values are in the array. This is how many lasers there are being picked up by the xbox kinect.
- ☐ Laser Data at Position is a little more interesting. It will get the element at the position you give it, from the array.
- ☐ Arrays start counting at 0. So the first element of the Laser Array is element 0. The final element of the array is Laser Array Length - 1 element.
- ☐ The minus one is because we're starting at 0!

🚩 Test the values

✓ Activity Checklist

- ☐ As we did before, we're just going to get a better understanding of these blocks by double clicking them!
- ☐ Start with the Length block. Everyone using the robots should get the same value for this because they all use the same laser sensors!
- ☐ Next test out the Laser Data at position block.
- ☐ You can enter any number between 0 and the Laser Array Length.

Challenge: Re-create the Front Laser Distance block

Using the two new blocks we've learnt here, and some basic mathematics, can you work out how we can make our reporter for the Front Laser Distance?

Bear in mind that there are equal lasers either side of the center!

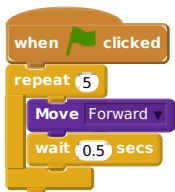
Step 2: More Complex Loops

✓ Activity Checklist

- ☐ We've learnt about loops, and we've learnt about the Laser Sensors. We're going to combine them to make a slightly more complex loop.
- ☐ We're going to use a new type of loop.



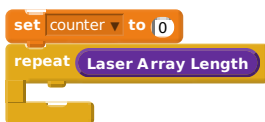
- ☐ This is extremely similar to the repeat until block that we've already seen. However, as you might have guessed it repeats a specified number of times.
- ☐ So the loop that performs 5 times that we learnt last time can be done with this. Let's quickly see that to remind us of loops.



- ☐ This will act exactly like the one we did before.
- ☐ We want a loop, that will visit every element of the Laser Data Array. Now this is quite complex, if you think you understand please do play with it and test your ideas.
- ☐ If you don't quite understand, that's expected! This is some tricky stuff.
- ☐ We'll start off by using a variable that will keep track of our current element, so make a variable called counter, with an initial value of 0.



- ☐ After this we want a loop that will run as many times as we have items in our list of laser sensors.
- ☐ Remember, we can use some of our new blocks as numbers in loops!

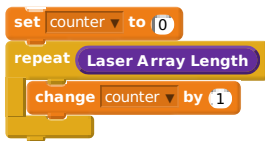


- ☐ Excellent! We've got a loop that will happen the amount of times that we have elements in our Array.
- ☐ We don't have to -1 because it's repeating that many times. If we want to get the last element, however we have to ask for the length-1 element.

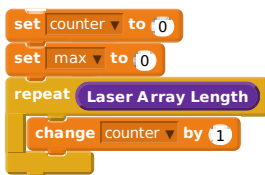
Step 2: Find the greatest distance

✓ Activity Checklist

- ☐ The next step is to increase the counter each time it loops, if we want to reference the current element of the Laser array, we need to know which one we're referring to.



- ☐ So this still won't actually do anything on the robot, but we're one step closer to what we want.
- ☐ From here, the goal is to find out which of the values of all of the laser sensors is highest (greatest distance).
- ☐ So, the first thing to do, is create another variable, which will store the currently highest value that we've seen. Call the variable max and set it to 0 to begin with.



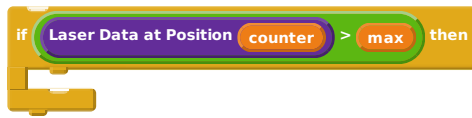
- ☐ Now we need to use an if block, just like we did before.
- ☐ What do you think will go in our if block?
- ☐ We want to check if the current item in the array, is higher than the current maximum that we've seen.
- ☐ First let's think about how to get the current element of the list.



- ☐ Okay so this block is the value of the item we're currently looking at. Now we need to put it into an if block that checks if this is greater than the current highest.



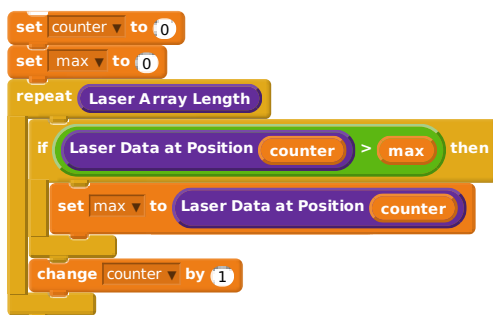
- ☐ This is the calculation we want right? Do you understand why?
- ☐ Next let's put this into an if block so that if it's the case we perform something inside of this loop.



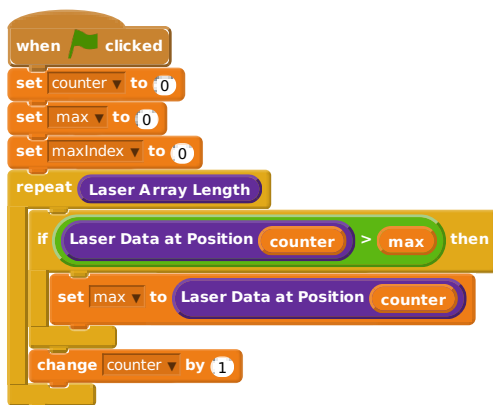
- ☐ Excellent! Now what do we want if this is the case, keeping in mind that this means we have a new maximum value.
- ☐ We should make the maximum value to this new value right?



- ☐ Fantastic, now we want to do this for every element of the Array, so this can go straight up into our loop!



- ☐ Note that we put the counter increase after the if block. This is because we want to finish what we're doing with the current element before increasing it!
- ☐ So to run it, we need to add a way to run it!



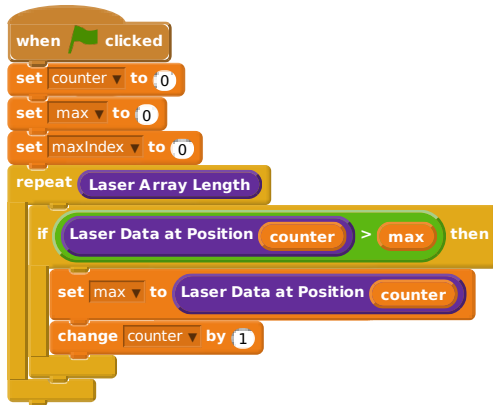
- ☐ Once it's run, you can double click max, to see what the maximum is!

Step 3 : Which laser though?

✓ Activity Checklist

- ☐ Do you see a problem yet?

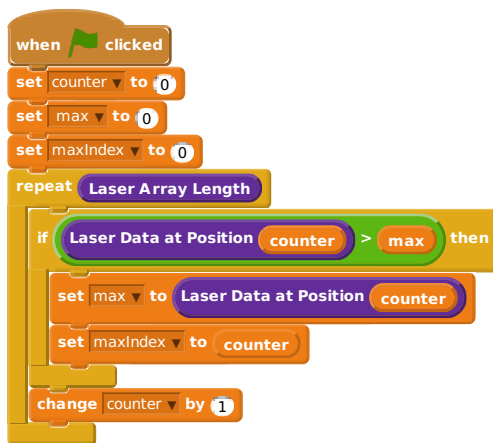
- ☐ We know the value of the laser with the furthest distance, but we have no idea where abouts in array it is!
- ☐ We're going to have to add a new variable.
- ☐ Make a new variable called maxIndex, this will hold the current index of the highest value, in addition to the variable that holds the value of it.
- ☐ Before your loop set maxIndex to 0, like with counter and max.



- ☐ Now can you figure out how we update this? We also only have to update this variable if we're updating the maximum value, so we can use the if block.



- ☐ This is what we want to perform, see if you can guess where in the code it goes.



Test your Project

Activity Checklist

- ☐ When you run it take a look at the maxIndex value, remember that anything lower than the middle of the list is to the left of the centre and anything higher than the middle is to the right.
- ☐ Try and predict what range the laser will be in when you face it towards/away from certain objects.

- ☐ The corner of a room is a great way to test something like this.

Challenge Repeat until loop

Can you re-create this cod but using the loops we learnt last session instead? This would be the repeat until block.

```
```blocks
repeat until ()
```
```

This is quite tricky but would be very impressive, and also fantastic practice.

Step 4: Using this to do something

✓ Activity Checklist

- ☐ Now we can identify the location in the array for the largest distance. But we're not doing anything with it, right?
- ☐ We're going to make the robot rotate towards greatest distance.
- ☐ Using the code from before, we want it to decide whether the index that the loop gives us is to the left of the centre, to the right of the centre or in the centre exactly.
- ☐ We've seen if statements before. Before we put them into the main program try and work on the if statements, so that they're ready to put right in.
- ☐ Remember that anything lower than the centre is to the left, and anything higher is to the right. The sensor values go from the left to the right.

`maxIndex > Laser Array Length / 2`

`maxIndex < Laser Array Length / 2`

- ☐ Note: The length of the array divided by two is the midpoint!
- ☐ These are the two operators that give us the information we want, the Yes/No questions we need for the if statements.

```
if maxIndex > Laser Array Length / 2 then
  [ ]
if maxIndex < Laser Array Length / 2 then
  [ ]
```

- ☐ The first if block is triggered when the highest value is to the right of the centre (Greater than)
- ☐ The second if block is triggered when the highest value is to the left of the centre (Lower than)
- ☐ Now we can implement our move blocks!
- ☐ We want it to rotate towards the highest value its found, so rotate left for anything on the left, and right for anything on the right.

```
if maxIndex > Laser Array Length / 2 then
  Rotate Right
else
  Rotate Left
```

- ☐ That should do it! Let's add this to our main program.
- ☐ Keep in mind we don't want to do this in the loop, we need to perform this AFTER we've finished deciding on the greatest distance.

```
when clicked
  set counter to 0
  set max to 0
  set maxIndex to 0
  repeat Laser Array Length
    if Laser Data at Position counter > max then
      set max to Laser Data at Position counter
      set maxIndex to counter
    change counter by 1
  if maxIndex > Laser Array Length / 2 then
    Rotate Right
  if maxIndex < Laser Array Length / 2 then
    Rotate Left
```

Test your project

Activity Checklist

- ☐ We've worked hard to get this far!
- ☐ Clicking the green flag will run your code.
- ☐ Does it rotate?
- ☐ Yep, but only once!
- ☐ Let's work on that in the next exercise.

Save your project

