Extra Boolean Operators

Introduction

We've used a lot of conditions in previous exercises, those are the things inside of if blocks. We're going to look at how we can group these together, using some more operators on Scratch!

Step 1: The blocks

Activity Checklist

- [ ] Let's start by getting familiar with the types of block we'll be using.

- [ ] If you can remember, we described condition blocks as Yes/No questions. So in an if block it decides whether to proceed based on the answer.

- [ ] These still fit in with these new blocks we have. It's quite intuitive too.

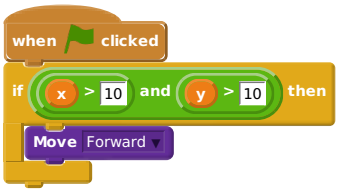
- [ ] For instance, the and block, you have to give it to of these Yes/No Questions. The and block itself will produce a Yes/No. It will only say Yes if both of the blocks reply with Yes, as the name dictates it's the first one AND the second one.

- [ ] Similar to above the or block will only produce Yes if ONE or both of the blocks reply yes, it requires one OR the other one.

- [ ] The Not block, is slightly different, it only takes one item. It will just flip the answer. So if the answer is Yes, it will swap it to no, and vice versa.

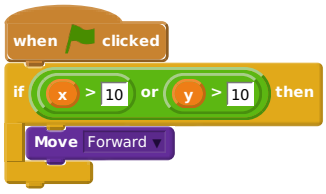
- [ ] Let's make two variables, X and Y, and assign random numbers between 1 and 20.

- [ ] Great - Now we'll try out the different blocks that we've seen.

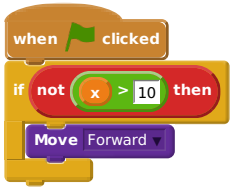


- [ ] Now this functionality isn't anything particularly useful, but it's showing you how the block works. Try it with the or block,

too.



- [ ] One last thing we'll try the not block.



Test your project

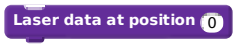
Activity Checklist

- [ ] We've made some structures with these new blocks.
- [ ] At the moment the behaviour will be unpredictable, it relies on random numbers!
- [ ] At this stage it will be useful to use non-random x and y values and predict whether the Turtlebot will move forward or not
- [ ] Don't be afraid to move things around!

Step 2: Make it more useful

Activity Checklist

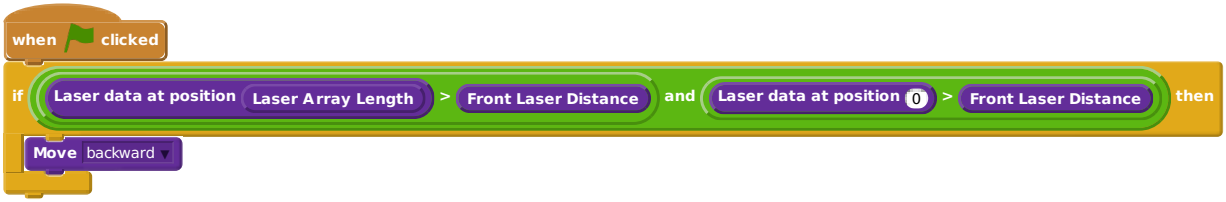
- [ ] I hope you're more confident with these new blocks by now, we're going to put them into practice.
- [ ] We'll use our laser data sensor. We will want information about the furthest left and furthest right laser.
- [ ] If you've been paying attention you should be able to guess that will be Laser Data at position 0, and at position (Length - 1)
- [ ] It's not 1 and Length because the array starts at 0! So we minus one from both of these.



- [ ] We want to move backwards if both the Left laser and the Right laser have a value greater than the middle laser!
- [ ] First we'll make blocks for these two, using the > block.

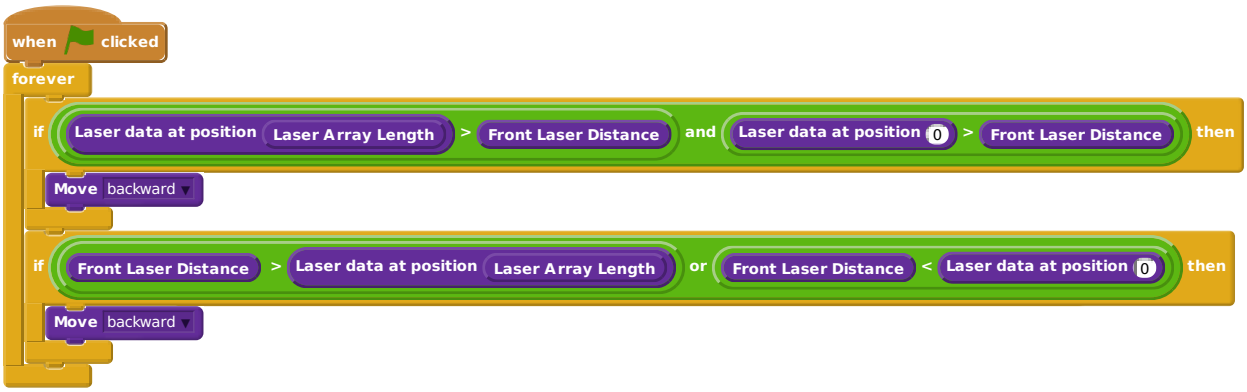
- [ ] Do we want an and, or an or?
- [ ] An and, right? We want both of these things to be true.

- [ ] As before let's put this into an if block! And if it's true, move backwards.



- [ ] Excellent, Test it out! It won't ever rotate, and if it's going to move, it should only do it once.
- [ ] You probably guessed, we want to put this in a forever loop so it keeps going!

- [ ] Again, at some point it will stop because once the front laser distance is not lower than the other two, it has nothing to perform.
- [ ] Make it so that, after the above, if the front laser distance is lower than the left, or lower than the right, it rotates left.



- [ ] That's a lot of blocks! Make sure to play around with it and test things out before moving on.

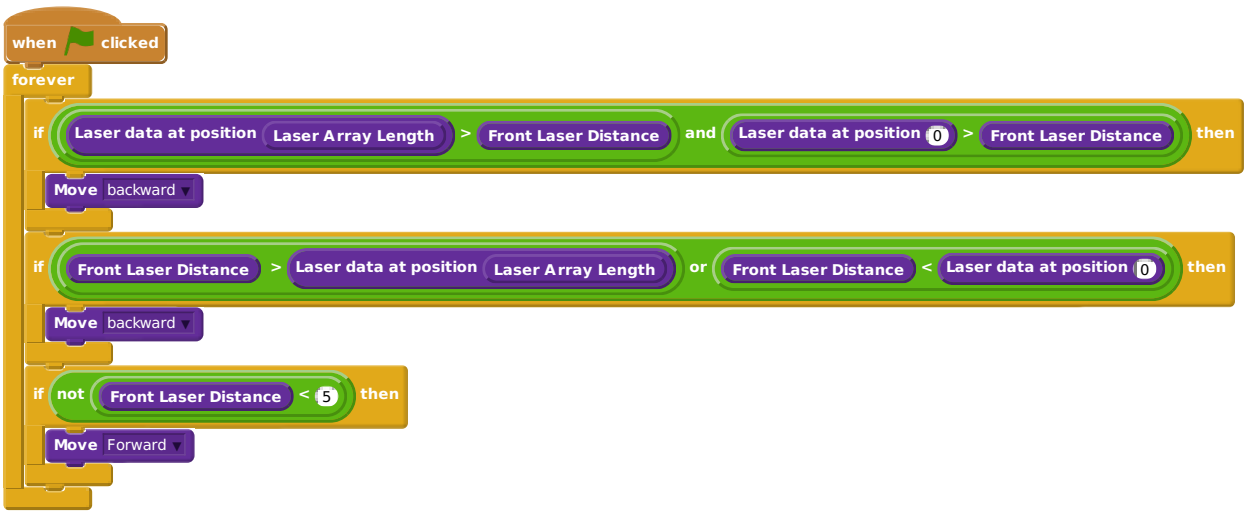
Step 3: One last block, not!

Activity Checklist

- [ ] The only thing we're missing now is the not block.
- [ ] We'll have a new if, below the current two, that says if the front laster distance is not below 5 we'll move forwards.
- [ ] Let's start by creating the condition.

```
 < not <(Front Laser Distance::custom) < (5)>>
```

- [ ] The not block can make it confusing, but read them seperately.
- [ ] The bit inside of the not block will return Yes if the distance is lower than 5.
- [ ] The not swaps this to a no!
- [ ] Now let's add the if block.



- [ ] Set yourself some of your own tasks, what conditions can you combine to produce something interesting?

Save your project