

Explore a Room

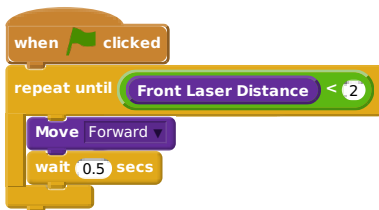
Introduction

We've used loops, we've used movement, and we've used laser sensors. This time we're going to use all of these things with the addition of random numbers! Random numbers are used commonly in programming so it's really useful to understand them and be able to use them as you wish. Without further ado let's get into it.

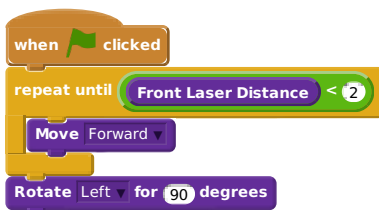
Step 1: Revisit Exercise 3

✓ Activity Checklist

- ☐ Let's load the code that we wrote in Exercise 3.



- ☐ So this code will move until it's 2 meters away from an object.
- ☐ Using our new knowledge of forever loops we can make it repeat this.
- ☐ However, if this alone was in a forever loop it would just stop when it would without the loop because the front laser distance will stay below 2!
- ☐ So before we put it in a loop let's make it turn when we get within 2 meters of an object.
- ☐ For this we should use the turn an amount of degrees block. Start with just turning left 90 degrees.



🚩 Test your project

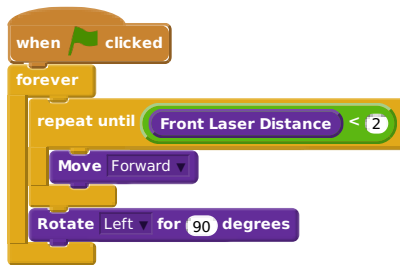
✓ Activity Checklist

- ☐ Press the green flag and observe what happens.
- ☐ It will do exactly what it did before except now it will rotate left once it's reached 2 meters distance from a wall.

Step 2: Now let's loop it!

✓ Activity Checklist

- ☐ We want to repeat this forever, we've used forever loops before, now it's time to put them to practice!
- ☐ Try and add the loop before looking at the next section to check you understand it!



- ☐ Don't forget to implement our interrupt, like we made last session, we're using a forever loop afterall!



🚩 Test the loop

✓ Activity Checklist

- ☐ Again, if you hit the green flag you can observe what the program does.
- ☐ It will essentially start following the wall around the room eventually.
- ☐ It's a great opportunity to try out different values as well as possibly rotating right aswell.

Step 3: Randomness

✓ Activity Checklist

- ☐ We've not seen any random numbers yet, so we'll start off basic.
- ☐ If you navigage to the Operators tab you should see a block for picking a random number.



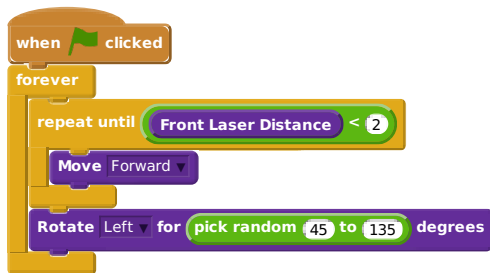
- ☐ I'm sure you can predict what this will do; this is a number we can use in our code as numbers, like variables. It will pick a

random number between the two values you give it, including the two values.

- ☐ Let's use this to rotate a random amount, rather than 90 degrees. This means we can explore a room, rather than follow a wall.
- ☐ You could use numbers between 45 and 135 for a good range around turning at a right angle, but this values really don't matter too much, providing you understand what it's doing.

Rotate Left for pick random 45 to 135 degrees

- ☐ Here's the rotation block; let's just plug it into the program we have so far.



Test your program

✓ Activity Checklist

- ☐ Let's run what we've made so far!
- ☐ It should start exploring the room, Brilliant!

Step 4: More Randomness

✓ Activity Checklist

- ☐ So we've got something random, however we can't help feeling like it's mostly going towards it's left.
- ☐ That's because it is!
- ☐ We could fix this by increasing the range of the random number, but it'd be better if we implemented it randomly choosing between left and right, too.
- ☐ How can we pick at random between rotating left and right?
- ☐ Well, we already know how to generate a random number.
- ☐ We also already know how to use if statements!
- ☐ How can we make it do one thing half of the time, and the other thing the other half of the time?

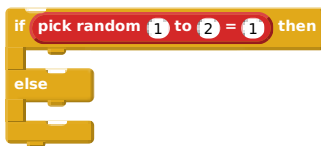
- ☐ Well we can start by generating a random number between 1 and 2.

A green Scratch block with the text "pick random 1 to 2".

- ☐ The values 1 and 2 will both appear equally (over time!)
- ☐ So we can use them with an if statement.
- ☐ For this you could set the random number generated to a variable and then check if it's equal to 1 in one if block, and check if it's equal to 2 in another if block.
- ☐ Instead, I'm going to show you how to use a new control block, the if else block.



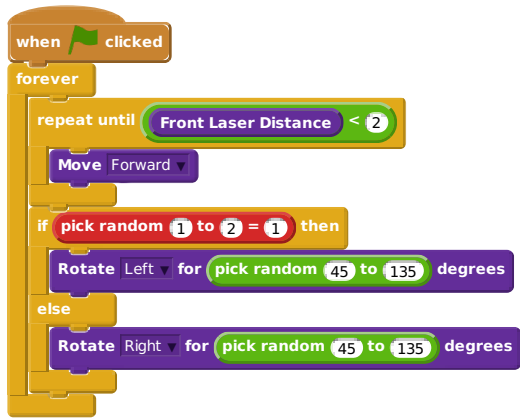
- ☐ The way this works is similar to before. The if then part, will work exactly how it did before, when the item in this block tells us Yes, it will perform the blocks we put into the top half of this block.
- ☐ The Else part of the block will be executed when the if then part returns No. So as the name of the block suggests, if something is true do this, else do something different.
- ☐ We can use this here, if a random number between 1 and 2 is equal to 1, rotate left, otherwise rotate right.



- ☐ Now let's add the rotation into the mix



- ☐ This is precisely what we want. This code it's will rotate left/right at random.
- ☐ You can add a hat block onto this, so that it runs when you click the green flag, and watch this in action.
- ☐ Now we need to replace our old rotation in the main program, with this new improved random rotation!



Test your project

Activity Checklist

- ☐ We've made some great progress, and re-used some of our older programs.
- ☐ Now we can spend some time testing out what we've developed, and making sure it's what we want.
- ☐ If you haven't already run it
- ☐ It should explore with a bit more randomness now!

Save your project