

Senn Robotics Team

Module 06: Line Following

While Loops

This program builds on the sensor work you did in the previous module. As you may have noticed in the last module, stacking up a series of if-else statements can get sort of clumsy. C provides a handy logical structure to allow you do actions while some action is true. It's called a while() loop.

```
while (condition to check) {  
    list of statements to do each time through the loop  
    at the end of the list, go back and check if the condition is still true, if so  
loop again, if not, break out of the loop and proceed along  
}
```

Here some example of conditions:

```
i > 4 (greater than)  
i < 10 (less than)  
i >= 512 (greater than or equal to)  
i <= 512 (less than or equal to)  
i == 1 (exactly equal to. BE CAREFUL TO USE DOUBLE == and not =)  
i != 1 (not equal to)
```

Exercises

1. Take a moment to review all the variables that are set up and how they are used. You will only need to change certain items once.
2. The first loop keeps the robot moving. Run the robot a few times. Does it always stop on the black line? Why or why not?
3. Modify the code so that the robot will always stop on the black line.

Exercises

Write code that does the following:

Line following is an important strategy that helps move you away from relying in the imprecision of dead reckoning. With line following, you try to detect a line and then follow it to known landmarks. If you position the robot anywhere on the game board and aim it generally toward a black line, it will go until it reaches the black line. Once it reaches the black line, it follows it indefinitely. The general strategy is to check to see if you're on black. If not, turn in one direction until you hit black and then move forward.