

While Loops Explained

Author #1 William Pfeffer

Assignment Definition And General Feedback By Michael Panitz
at Cascadia Community College (<http://www.cascadia.edu>)

Table of contents:

- [Quick Summary](#) *
- [When To Use \(or Avoid\) This](#) *
- [Example Of Usage](#) *
- [Example Trace](#) *
- [Syntax Explanation](#) *
- [Help With The Logic](#)
- [Important Details](#)
- [Common Mistakes \(And How To Fix Them\)](#)
- [Hyperlinks \(if any\)](#)

(Required sections have a * after them in the above list)

Quick Summary:

In Java, you can use while loops to repeat a command as long as the statement listed in it is true. For example, if you want to move a robot forward as long as the font is clear, you would use the code listed in the example code. The main purpose of a while loop is to save time by having Java loop through a line of code until it comes out false.

When To Use This / Avoid This:

There are several situations where you will want to use this technique. Generally, you use while loops to eliminate the need to use a command multiple times. This, for example, you could use if you want to move a robot named robot1 to the avenue that robot2 is located on at that time. You could use while loops to accomplish the task of moving robot1 to the same avenue as robot2. While loops can be used in many different situations to save time overall. It's also good to note they can be put into a new command to be called on later if you plan on using it multiple times.

There are several situations where you will want to avoid this technique. In some cases, adding a while loop to your code will produce an undesired outcome. In some circumstances you don't want to use a while loop, and rather and if statement. In some situations, while loops will cause a command to be repeated an infinite amount of times, which is undesirable in most situations. To avoid this you must first assess if a while loop is going to do exactly what you want the program to do.

Example Of Usage:

Let's say you want to make a robot that moves until it hits a wall no matter what distance there is between the starting position of the robot, and the wall. Assuming the wall is located on the same street as the robot, and that the robot is facing the direction of the wall, we would use the code listed below to get the robot from point A (Starting Position) to point B. Seeing how we can't determine the number of spaces between the robot and the wall, we need to use a while loop in order to get the robot to its destination.

There are 2 steps:

First, we need to create a new public void, and decide what we want to name the new public void. In this case I will name it moveforward().

Second, we write the while loop within the public void. With a while loop we need to make sure that the code is written correctly. Notice the parenthesis, brackets and the capitalization, as they are very important and cause compile errors if written incorrectly. The goal of this while loop is to make the robot move forward until he reaches a wall. In this case the wall is located 4 spaces away from the robots starting location, so it will loop over the command until it turns out to be false. Also, take note of the while loop located in main. This loop will repeat itself until there is no longer anything to pick up. In this case, there is nothing there to pick up from the start.

Line	Code
1.	<code>import becker.robots.*;</code>
2.	
3.	<code>class RobotExample extends Robot</code>
4.	<code>{</code>
5.	<code>RobotExample(City c, int st, int ave, Direction dir, int</code> <code>num)</code>
6.	<code>{</code>
7.	<code> super(c, st, ave, dir, num);</code>
8.	<code>}</code>
9.	
10.	<code>public void exampleRun()</code>
11.	<code>{</code>
12.	<code> while(frontIsClear())</code>
13.	<code> {</code>
14.	<code> this.move();</code>
15.	<code> }</code>
16.	<code>}</code>
17.	<code>}</code>
18.	<code>public class example extends Object</code>

19.	{
20.	public static void main(String[] args)
21.	{
22.	City woodinville = new City();
23.	RobotExample rob = new RobotExample(woodinville, 1, 1, Direction.EAST, 0);
24.	new Wall(woodinville, 1, 5, Direction.EAST);
25.	
26.	rob.exampleRun();
27.	while(rob.canPickThing())
28.	{
29.	rob.pickThing();
30.	}
31.	}
32.	}

Example Trace:

In a nutshell, this technique allows you to move the robot the correct distance assuming that the wall is placed at a random place.

In order to go over these details more thoroughly, here is a (partial) trace of the above program, with some additional explanation afterwards

Line #	Program Statement	Robot street #	Robot Ave #	robot Dir	robot Backpack	TRUE/FALSE	WALL
22	City woodinville = new City();				0		
23	RobotExample rob = new RobotExample(woodinville, 1, 1, Direction.EAST, 0);	1	1	E	0		
24	new Wall(woodinville, 1, 5, Direction.EAST);	1	1	E	0		(1,5,E)
26	rob.exampleRun();	1	1	E	0		(1,5,E)
10	public void exampleRun()	1	1	E	0		(1,5,E)
12	while(frontIsClear())	1	1	E	0	TRUE	(1,5,E)

14	this.move();	1	2	E	0		(1,5,E)
33	while(frontIsClear())	1	2	E	0	TRUE	(1,5,E)
34	this.move();	1	3	E	0		(1,5,E)
10	while(frontIsClear())	1	3	E	0	TRUE	(1,5,E)
12	this.move();	1	4	E	0		(1,5,E)
15	while(frontIsClear())	1	4	E	0	TRUE	(1,5,E)
17	this.move();	1	5	E	0		(1,5,E)
18	while(frontIsClear())	1	5	E	0	FALSE	(1,5,E)
27	while(canPickThing())	1	5	E	0	FALSE	(1,5,E)

The program starts at beginning of the **main** function, on line 22, like normal. After the walls and robots load, it then proceeds to go to line 26 which calls upon the command that we made on line 10. It then proceeds to loop over the while loop until it's no longer true.

Syntax Explanation:

Note on syntax of command: You can use while loops in a number of different ways. You could for example use a number of commands to loop over and over again until its no longer true. In this case, the while loop is very basic. It simply keeps the robot moving as long as the path is clear in front of it.

Below, you can see the finished program with a while loop that does the same thing as the code in example of usage. Note the highlighted material, because its what we want to focus on right now. Its important to have correct syntax when writing any while loop. The most important things to note are:

1. You **MUST** have open and closing braces! Failure to put these into the code will more than likely result in a compile time error. The exception to the rule being that if you are only listing one command after the loop it will still compile, but its always recommended that you put these here just as a good habit.
2. Case sensitivity is a very important aspect to take into account with working with the Java language. For example if you were trying to get a robot named joe to move, and you accidentally typed Joe, Java would not be able to find the robot, and would result in a compile time error.

Line #	Program Source Code
1.	import becker.robots.*;
2.	class RobotExample extends Robot
3.	{
4.	RobotExample(City c, int st, int ave, Direction dir, int num)
5.	{
6.	super(c, st, ave, dir, num);
7.	}

8.	public void gogo()
9.	{
10.	while(frontIsClear())
11.	{
12.	this.move();
13.	}
14.	}
15.	public void turnRight()
16.	{
17.	this.turnLeft();
18.	this.turnLeft();
19.	this.turnLeft();
20.	}
21.	}
22.	public class Example extends Object
23.	{
24.	public static void main(String[] args)
25.	{
26.	City woodinville = new City();
27.	RobotExample rob = new
28.	RobotExample(woodinville, 1, 1, Direction.EAST, 0);
29.	new Wall(woodinville, 1, 5, Direction.EAST);
30.	new Wall(woodinville, 5, 5, Direction.SOUTH);
31.	rob.gogo();
32.	rob.turnRight();
33.	rob.gogo();
34.	}
35.	}

The syntax of a while loop is fairly simple. The general syntax for writing a while loop looks something like this...

```
while(robot1.getAvenue() < robot2.getAvenue())
{
    <robot>(or <this> if used in public void).move();
}
```

You can substitute in a number of different commands within the parenthesis, for example this.getAvenue() !=5 would carry out the command as long as the avenue isn't equal to five. You can do the same thing with robot.getStreet() !=5.

Help With The Logic:

This is good to use whenever you have a circumstance in which there is a random distance between the robot and a point. Another solid use of this is to get a robot to another robots location, or to the location of another object such as a thing or a wall.

Important Details:

- It's important to remember that when using a while loop, or anything in Java for that matter everything is case sensitive, also you need brackets and parenthesis. If you forgot to do either of these things it could, and will result in the program not being able to compile.
- While loops will keep repeating the command until it is no longer true, so for some situations its best to use other types of commands.

Licensing



While Loops Explained by William C Pfeffer is licensed under a [Creative Commons Attribution-NonCommercial 3.0 United States License](https://creativecommons.org/licenses/by-nc/3.0/).

Plagiarism

If you believe that some or all of this document infringes on your intellectual property (i.e., part or all of this document is copied from something you've written) please immediately contact Mike Panitz at Cascadia Community College (perhaps using the Faculty And Staff Directory at <http://www.cascadia.edu/pages/searchtemplate.aspx>)