1001ICT Introduction To Programming 2015-2
Laboratory 7

School of Information and Communication Technology
Griffith University
September 10, 2015

<table>
<thead>
<tr>
<th>When</th>
<th>Teaching week 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goals</td>
<td>In this laboratory you will write programs with loops, type casts, and methods.</td>
</tr>
<tr>
<td>Marks</td>
<td>5</td>
</tr>
<tr>
<td>Robots</td>
<td>Cyclops-NXT</td>
</tr>
<tr>
<td>Props</td>
<td>Bollard</td>
</tr>
<tr>
<td>Track</td>
<td>WhiteBlack Track</td>
</tr>
</tbody>
</table>

1 Preparation

Before your lab class:

- Print these lab notes. You need to refer to them a lot before the lab class and during it.
- Read up to section 17 of the lecture notes.
- Browse the console and nxt environment documentation available at http://www.ict.griffith.edu.au/arock/itp/students/mash/.
- You can start work before your lab class. If you can’t write the complete programs, you could at least create the program files, with header comments, imports, and main method.

2 Pre-laboratory questions (0.5 marks)

Answer the following questions in the space provided, before your laboratory class.

1. (a) How many motors does the Cyclops robot have? __________
   (b) What do you need to do to make this robot move forward in a straight line?

   

   (c) What do you need to do to make this robot turn?

   

2. (a) What method returns a random value? __________
   (b) What type does it return? __________
(c) What, in your own words, is the difference between coercion and casting?

(d) To force a narrowing conversion explicitly, do you use coercion or casting? _______

3. If a method is a function, what kind of statement must it contain? _______

3 Activities

All programs must:

- have header comments showing the name of the file, the author’s name, and the purpose of the program;
- be written with at least a main method; and
- use constants for motor and sensor ports;
- be neatly indented; and
- use either style of bracing, being consistent throughout the program.

3.1 MaSH nxt program 1 (1.5 marks)

- Write a program that drives the robot forward on the track. The robot should stop when either the robot runs into an obstacle (the bollard) or drives on to the dark area.
- Hint: You can’t use waitForDarker and waitForPush at the same time. You will have to write your own loop that waits for either change.

3.2 MaSH nxt program 2 (1.5 marks)

- Write a program that makes the robot keep moving, backing off and turning away from obstacles as in this movie.
- This function uses that to return a random integer \( i \) such that \( a \leq i \leq b \).

```java
// iRandom(a, b) returns a random integer between and inclusive of a and b.
// precondition: a < b
int iRandom(int a, int b) {
    return (int) (random() * (b - a + 1) + a);
}
```

Use it to generate random time intervals during which the robot reverses and turns.

3.3 MaSH console program 1 (1.5 marks)

- A DNA sequence is a long chain of the bases, guanine (G), adenine (A), thymine (T), and cytosine (C).
- Write a program that prints out a random sequence of 1000 bases, like this:

```
$ mash DNA
mashc DNA.mash
javac DNA.java
java DNA
AGCAAAAGCAAGGATCCAGAAGTAGTGATATGAAACAGGCGCCTAGGTAAAGAGCCAGTCGTTGACGTGACACCCGTTGT
```
3.4 MaSH console program 2 (no marks, just kudos)

- Write function that computes the integer binary logarithm of a positive whole number.
- The integer binary logarithm of a positive integral number may variously be defined as:
  - the position of the most significant (left-most) bit equal to 1 in the binary representation of the number;
  - the largest $n$ such that $2^n$ is not bigger than the number;
  - How many times you need to divide the number by 2 to get 1.

3.5 MaSH console program 3 (no marks, just kudos)

- Write a program that reads an integer $n$, and keeps flipping a simulated coin until it flips $n$ heads in a row, then prints the total number of flips it required.

3.6 MaSH console program 4 (no marks, just kudos)

- Write a function that simulates rolling a die (like we did in a workshop).
- Write a program that tests how fair your die function is, by running it many (millions!) of times and printing the percentage of times each number results.
- This will be easier with arrays, but possible without.

4 After the Laboratory

- Organise the work you have done into folders on your network drive.
- Please answer these feedback questions.
  - What was the most difficult aspect of this laboratory?
  - Did you find an error in these lab notes?