Institute of Technology Blanchardstown

Algorithmic Problem Solving Project:

Otto the Robot

Dr. Kevin Farrell
13 October 2016

1 Contents

2 Acknowledgements/Credit

This assignment is adapted from previous assignments created by Lecturers Mark Cummins and Noel Carey.

3 Objectives

The objectives & Deliverables for the project are as follows:

1. Work in teams of four students (max.). All team members **<u>must</u>** belong to the same practical group.
2. Read the background story and instruction set. Discuss each of the instruction set commands, assumptions and rules to ensure you understand everything. (If you're not sure of something, please ask the Lecturer)
3. Complete each of the tasks detailed below, and document your work. You may find it helpful to use multiple approaches, and in some cases, trial and error, to solve the more complex tasks.

4 Deliverables

1. Write a report of no more than 3,000 words, describing how you solved the tasks and providing the actual solutions. Your report must contain:
    a. A list of **<u>every member</u>** on your team (student number and full name) on the title page.
    b. Your team's solution to each task: Please type out the commands. **<u>Do not</u>** include them as a screen shot.
    c. A description of how your team approached each task: what you tried? What worked or didn't work, and anything you learned from attempting the particular task.
2. Name your report-file in accordance with the following naming rule: Surname1_Surname2_Surname3_Surname4_APS.docx
3. **<u>One member only</u>** of each team must upload the team's report in **<u>M$ Word or LibreOffice ODT format</u>** to the Turnitin link on moodle. This is a **<u>one-time only</u>** upload. You will **<u>not</u>** be permitted to upload a "draft" version and then a "final" version later. This will be strictly enforced. So, make sure all members of your team are happy with your report before you upload it. The deadline for uploads is the beginning of your practical session in Week 12; i.e. your practical session in the week beginning 5 December 2016.

4. Give a Presentation/Demonstration on your project in the practical session in Week 12.

5 Grading

The Project Report and Presentation/Demonstration are each worth 30 percentage points, giving a total 60 percentage points allocation to the module grade.

6 Background

You recently graduated with a programming degree, and by some amazing stroke of luck were hired by NASA to maintain their website and database. Barely two weeks after you were hired, you have suddenly received an urgent call from the head of the Space Exploration Division. Apparently, they are experiencing an unexpected problem with one of their deep space probes and their in-house programmer simply never showed up for work. They need you upstairs immediately…

You rush to the sixth floor, passing through several security checkpoints, and arrive at the spacecraft control room to a scene of considerable disarray. You survey the massive room, overwhelmed by the dozens of rapidly flashing monitors and the sense of near panic generated by several men in white coats scurrying about the control panels, shouting excitedly to each other. You stand at the entrance to the room for several minutes, fascinated and unsure where you fit into this drama, when you notice that one of the scientists has stopped moving about and is looking directly at you. After a few moments he abruptly starts walking in your direction.

Before you get a chance to speak he asks if you are the programmer, but before you get a chance to reply he grabs your arm and briskly leads you towards the row of control panels in the centre of the room. As you near the massive mainframe computer with its numerous panels of blinking buttons and assortment of flashing screens you notice that the other scientists have stopped talking and have all begun to move towards you. It gets so quiet after a moment that all you can hear is the low, steady hum of the cooling fans in the big computer. Standing amidst the dozen or so scientists in their white lab coats, you suddenly feel out of place in blue jeans and faded t-shirt.

The silence seems to drag on for an eternity, with all the scientists looking expectantly at you, when the man who led you to the control panel finally speaks. He explains that they are currently monitoring a little known deep space probe named ‘Magellan 2’, launched in 1972 to explore the moons of Saturn. The probe has finally reached Saturn, but apparently they made a slight calculation error in the original flight plan and they need to fire the aft directional rockets to correct its course.

The problem he elaborates, is that their sensors indicate some sort of obstruction in the fuel channels, (perhaps space dust accumulated over the years), and if they fire the rockets it would likely cause an explosion. However, if they are unable to correct the probe’s course it will definitely be pulled into Saturn’s gravitational field and meet a fiery end – obliterating years of research. Their one and only hope he says, rests on a small fail-safe built into the probe in the form of a micro robot called OTTO, designed to navigate and clear the probe's fuel channel. Unfortunately, all of the schematics for the probe got mixed up with the prototype so that NASA are not even sure about the exact shape of the fuel channel.

All they can determine, as a common characteristic of all the schematics is that the fuel channel has one entrance, one exit, no dead ends, and no intersections. Due to the compact and intricate nature of the probe's design, if OTTO stops anywhere other than on his recharge station it will become an obstruction itself and surely guarantee the demise of the probe. In addition, once OTTO leaves the recharge station it's battery will last approximately seven minutes, and it takes 38 minutes to transmit new instructions, so it would not be possible to transmit new instructions before the battery runs out should OTTO fail to return to the recharge station. The net effect of all of this is that you have one and only one shot at this…

You look around at the sombre, hopeful faces of the researchers, take a deep breath and begin jotting down algorithms…

7 Otto Instruction-Set, Assumptions and Rules

7.1 Otto Instruction Set

Otto only understands a few commands—its **Instruction-Set**—as listed below:

```
[1]  Stand up
[2]  Sit down
[3]  Take a step (must be standing)
[4]  Turn (right turn only; i.e. turn clockwise/–90 degrees)
[5]  Raise arms (must be standing)
[6]  Lower arms
[7]  Add one to memory
[8]  Subtract one from memory
[9]  Test: Touching anything? (must be standing with arms raised)
[10] Test: Touching a door? (must be standing with arms raised)
[11] Test: is number in memory zero
[12] Repeat [Instruction][x] times (for e.g.: 'Repeat turn 2
      times')
[13] GOTO (program will 'jump' to line number indicated; for e.g.:
      'GOTO line# 7' jumps to line 7)
[14] Open door (must be standing with arms raised)
[15] Stop (Otto will not shut down without a stop command and his
      battery will not recharge)
```

7.2 Assumptions

- OTTO will start in a seated positions with his arms lowered.
- Number in memory is zero

7.3 Important Rules

1. OTTO has a very intricate gear system driven by a single servo motor. If duplicate commands are received, such as an instruction to stand while OTTO is already standing, OTTO's delicate gears will become misaligned and OTTO will be disabled.

2. OTTO's limited instruction set DOES NOT allow for embedded/nested test statements.

3. OTTO must receive a 'STOP' command in order to be successful.

8 OTTO Program

You can test your solutions to each task by using the **Otto.exe** program, available on Moodle.

You can print out your steps using the Otto.exe program.

9 Task 1: Instruct OTTO to walk in a square three steps to every side making right turns only.

Using just the instruction-set, assumptions and rules given above, you need to come up with a set of instructions that will cause OTTO to walk in a square pattern, three steps on each side. Remember to check that you are following ALL of the assumptions and rules.

When you think you have come up with a correct solution, you need to test your algorithm.

One person should read out your instructions while another person should pretend to be OTTO the robot and perform the exact steps. You should probably also have one person checking at each step that none of the rules or assumptions are broken. If you spot any problems try to correct your algorithm and then test again.

When you are 100% sure you're algorithm is correct you should try to program OTTO the robot using the **Otto.exe** program on Moodle.

If you are successful then continue to task 2, if you experience an expected problem, correct and retest your algorithm before continuing onto task 2.

10 Task 2: Instruct OTTO to walk to a wall any distance forward and return to his recharge station. The wall may be any number of whole steps in front of OTTO.

This task is much more difficult and will involve lots of careful planning and testing. Remember the wall may be any distance away; your algorithm needs to work for all cases, so plan accordingly. Follow the same basic steps used to solve Task 1.

Once you've tried to solve this one on paper you can try testing it yourself using the **Otto.exe** file on Moodle, Use game mode or instruct mode to help you work your way through to a solution. Good luck.

11 Task 3: OTTO Solves a Maze

11.1 Introduction

This task is more difficult and complex than the other two. To solve this problem successfully you'll have to do a lot of planning first. You should try breaking the problem down into simpler steps, and into smaller more manageable pieces. This is a difficult challenge so you may not solve it completely, but you should still submit your best attempt. Good luck.

This task is broken into a number of sub-tasks.

Otto is set up to help you organise your solutions in a logical order and to realise that the computer cannot read your mind. You must give it a complete set of instructions. One of the most frustrating things when working with a computer is that it does exactly what it is instructed to do, not what you *thought* you instructed it to do.

11.2 Problem Descriptions

You need to instruct OTTO how to navigate through an **unknown** maze, so you need to give him a set of instructions that will allow him to find his way through **ANY** maze.

All you know is that all of the mazes are constructed with the same rules. So,

- The maze has no dead ends and no intersections.
- OTTO will enter and leave the maze at the same point.

Before you jump straight in and try solve this problem, we provide you with some simpler sub-tasks first that will hopefully help you with your final solution.

Remember, it's much easier to solve when you can visualise the problem, so have one of your group members act as OTTO and try visualising and testing the problem each time.

11.3 Maze Tasks Deliverables

For each of the four maze sub-tasks listed below, you should include in your report:

1. A description of your robot's logic, and your plan to solve the maze. An outline of the different ways you tried to solve the problem and any problems you encountered.

2. Your instructions to solve each step. Even if they don't work, you can include your logic in English, if you couldn't implement it with OTTO.
3. Your final Otto code. You can print out your steps using the Otto.exe program. Type the code into your report document.

Sub-Task 4 has an additional requirement, namely, that you must include a link to your online video (see below for details).

11.4 Sub-Task 1

How would you make OTTO turn left? Once you think you've figured it out, test it using one of the group as OTTO, and then try to solve the first practice maze (the 3*3 square) going in the opposite direction, so OTTO only turns left!

11.5 Sub-Task 2

You haven't used the open door and test door actions yet, so practise using them now. Using the hard level on the OTTO game, get OTTO to walk up to the door and open it, walk in, turn around and come back to the starting position. (The door is always 2 steps in front of OTTO).

11.6 Sub-Task 3

A slightly easier version of the full maze uses two doors; OTTO goes in one door and out the second door. From OTTO's starting position, the entrance to the maze is always two steps forward. When OTTO is facing the entrance, the exit is always 3 steps to his right. Figure 1 below shows two example mazes for you to test your solution. Try to solve this task before attempting the final task, using just the one door.

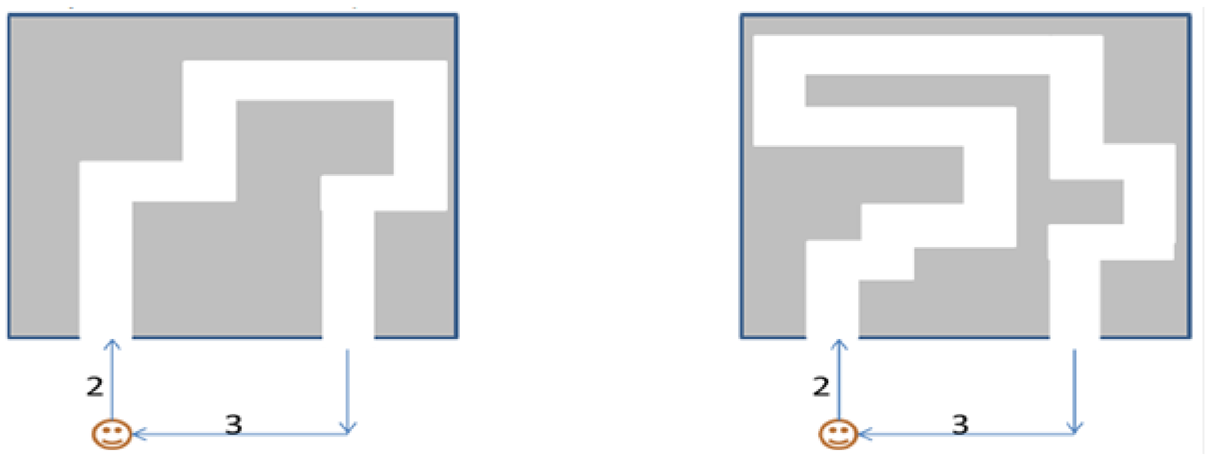

FIGURE 1: TEST MAZES FOR SUB-TASK 3.

Hint, you might want to count how many steps you take left and right…to know when you're at a possible exit location) This is the normal mode on the Otto Game.

11.7 Sub-Task 4

This is where you try to solve the full hard mode task for the OTTO game: a general maze with only one entrance and exit. If you've managed Sub-Task 3, it should be very similar. Good Luck.

You are required to video your team testing your OTTO maze solving algorithm. Even if your algorithm doesn't work you can still test it. You'll have to setup a maze on campus, using the buildings and corridors etc. you can even use stairs if you want. You should create fake barriers so that your maze follows the rules, i.e. No intersections, no dead ends etc.

> *For an example of the kind of thing I'm looking for see this video. http://www.youtube.com/watch?v=a9B9XJcMHw0*
>
> *You'll need to upload your video online somewhere, YouTube, vimeo etc. and include the link as part of your submission.*

12 Team Difficulties: Interpersonal Issues, Members not working, etc.

If there are difficulties with members in a team not getting along or not pulling their weight, please let me know **as early as possible** in the project so that we can meet to try to resolve the issues. Telling me during the 'deliverables week' will not enable a resolution.

13 Questions?

If you have questions or if some aspect of the project is unclear, please e-mail me, or ask me in class. E-mail: kevin.farrell@itb.ie

____

Ends.