Otto the Robot

Raigridas Bartkus, Timothy McCann, Benjamin Scallan

Leah Brennan [Group Seven].

B00094651, B00095944, B00097614, B00098851.

School of Informatics,

Department of Informatics and Engineering,

Institute of Technology,

Blanchardstown,

Dublin 15.

Word Limit for Assignment: 3,000. Total Word Count: 3,396

(Content: 1,244)

.

**Table of Contents**

Introduction……………………………………………………………………………………..3.

Part One: Otto Labs: Our Thought Process……………………...……………………………4-10.

Part Two: Finding Solutions …………………………………………………………...…...11-23.

Conclusion………………………………………………………………………………………24.

Index……………………………………………………………………………………………25.

**Introduction:**

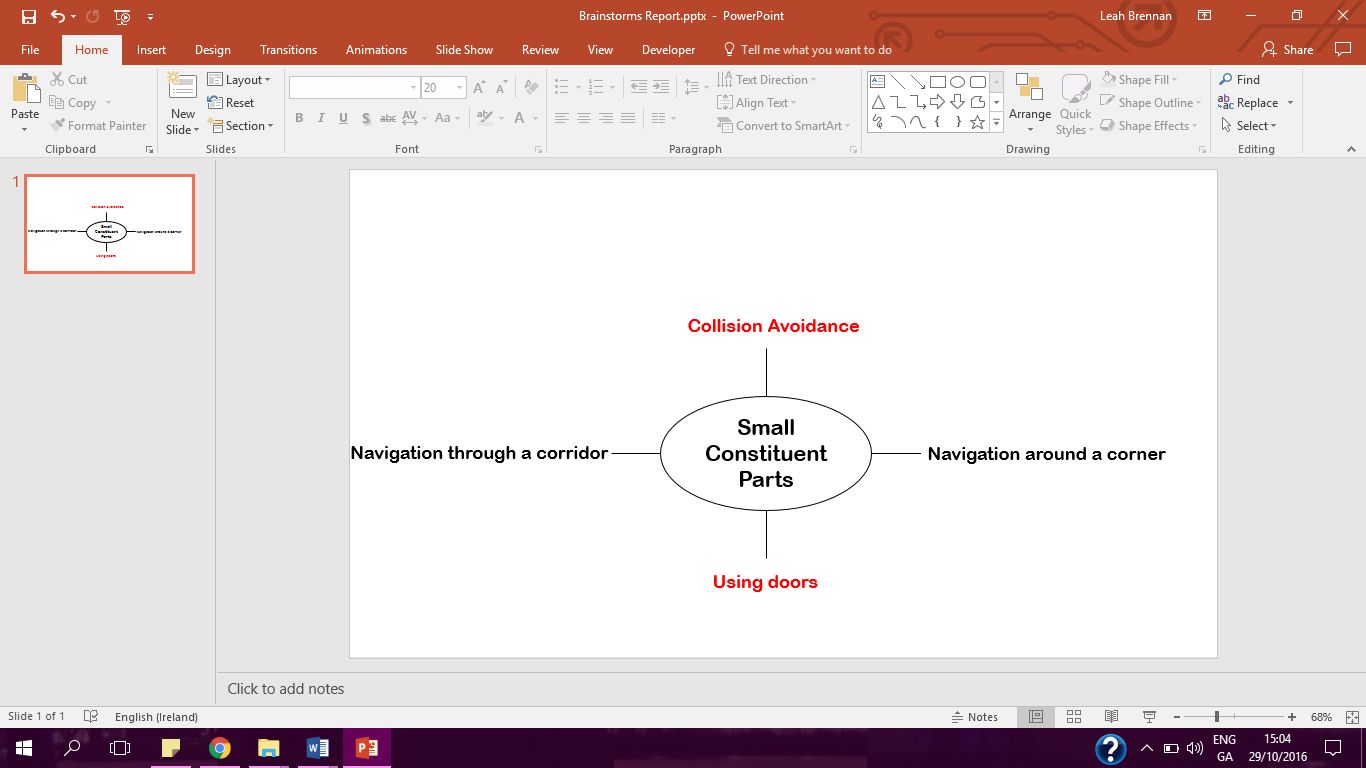
This was the first group project that we were assigned. We were relative strangers before this started so we didn’t know each other’s individual strengths. However, over the period of the project everyone’s separate talents were brought to bear enthusiastically. The completion of Otto the Robot has spanned over a month and a half and in that time we have gotten to know each other a lot better.

The robot’s intricate gear system and other limitations meant that Trial and Error would be our only option which also led to discussion among the whole of Group Seven. The advantages of working as a team were abundant, for example, each solution was reached together more quickly than they would have been individually, every team member had their own contribution which was helpful either in solving the tasks or writing this report and this co-operation has brought about the successful outcome contained in this report. This experience has been a useful lesson to us for our future in ITB and possibly in the workplace.

**Part One: The Otto Labs: Our Thought Process**

The first lab that involved this project was slow to begin with for the team. Following a rough delegation of roles, it was collectively decided to test every command available to us with an emphasis on the memory command. In this lab. Gerard O’Brien, along with his team members Victor Azzam, Damian Moore and Patrick Kerley shared one of their findings with the rest of the group; that Otto doesn’t recognize his charging pad which means he is also unable to know where he is in an unfamiliar maze. This highlighted the use of the touch anything test that gives Yes or No commands which cannot be repeated as a major factor in solving Tasks Two and Three.

Taking into consideration that a video of the team’s solution was to be filmed, the project was taken back to earth. It was suggested that the problem could be broken down into ‘its small constituent parts’ i.e. navigation around a corner, through a corridor etc. and by doing this each environment (or every task) is treated ‘individually’ and at the end of it put together to return the solution. We also jotted down that it is important to remember that it is an irregular shaped maze which means that more than just the most obvious problems needed to be solved. This is where the issues of collision avoidance and using doors to get Otto to his charging pad for the sake of NASA were added to what must be solved before a final decision was reached.

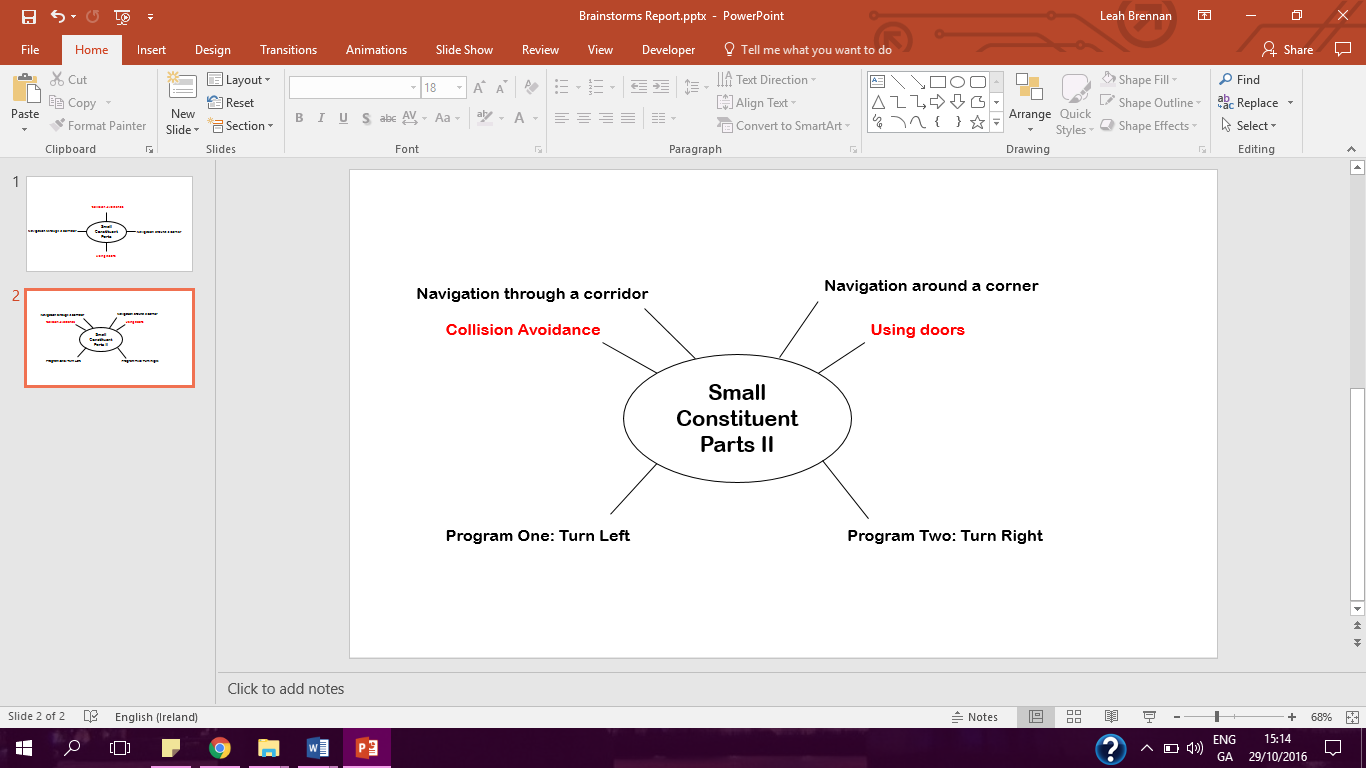


**Figure 1. Sample Brainstorm from Lab One.**

With Gerard’s team’s contribution and the team’s plan it was decided that the memory command which we were dealing with at the beginning of the lab was just an extra condition for specific instructions.

In the second lab, the team used the online program more than the first. It was a slow start again until we decided that the most difficult task should be solved first and then the commands to finish the easiest would be at our disposal. This would be the first step in how most of the lab was spent

We tried to delve deeper into Otto’s motivations for going into a maze and the robot’s definition of a door. We brought it back to the first lab, to the suggestion to break each maze down into its ‘small constituent parts’ this time we would devise one program in which Otto turns left and another where he turns right. As it states in the project description the starting procedure will always be ‘Stand up’, ‘Raise Arms’ and as we found out it the first lab; the ‘Touching anything?’ test. In one sentence, there should be a set of commands that will allow Otto to reach his destination.



**Figure 2. Additions to the sample brainstorm from Lab One.**

The team then spent the remainder of the lab tackling the most difficult task with the following sets of commands;

* i). 0) Start,

1) Stand up,

2) Raise arms,

3) Take a step,

4) Take a step,

5) Test: Touching Anything; Yes; Open door, No: Add one to memory,

6) Test; Touching Anything; Yes: Turn, No: Take a step,

7) Test; Touching a Door Yes: Go to Line 5, No; go to Line 6

* ii). 0) Start

1) Stand up,

2) Raise arms,

3) Take a step,

4) Take a step,

5) Test: Touching a Door: Yes: Open, No: Add one to memory,

6) Test: Touching Anything: Yes: Repeat turn 3 times, No: take a step,

7) Test: Touching a Door: Yes: Go to Line 5, No: Go to Line 6,

* iii). 0) Start

1) Stand up,

2) Raise arms,

3) Take a step,

4) Take a step,

5) Test; Touching Door: Yes: Open, No: Add one to memory,

6) Test; Touching Anything: Yes: Repeat turn 2 times, No: Take a step,

7) Test; Touching Door: Yes: Go to Line 5, No: Go to Line 6,

* iv). 0) Start,

1) Stand up,

2) Raise arms,

3) Take a step,

4) Take a step,

5) Test; Touching Door: Yes: Open, No: Add one to memory,

6) Test; Touching Door: Yes: Turn, No: add one to memory,

7) Test; Touching Anything: Yes: Turn, No: Take a step,

8) Test; Touching Anything: Yes: Go to line 6, No: Go to Line 7.

* v). 0) Start,

1) Stand up,

2) Raise arms,

3) Take a step,

4) Take a step,

5) Test; Touching Door; Yes; Open, No; Add one to memory,

6) Test; Touching Anything; Yes; Turn, No; Take a step,

7) Test; Touching Anything; Yes: Repeat turn 3 times, No; take a step,

8) Test; Touching Anything; Yes Go to Line 5, No: Take a step.

* vi).0) Start,

1) Stand up,

2) Raise arms,

3) Take a step,

4) Take a step,

5) Test; Touching Door: Yes: Open, No; Add one to memory,

6) Test; Touching Door; Yes: Turn, No: Add one to memory,

7) Test; Touching Anything; Yes: Turn, No: Take a step,

8) Test; Touching Anything: Yes: Repeat Turn 3 times, No: Take a step,

9) Test; Touching Anything: Yes: Go to Line 6, No: Go to Line 7.

**Figure 3. Trial and Error; Discarded Commands 1-6.**

The discarded commands of Figure Three show that this did not return positive results until we moved onto a process of ‘layering’ the commands rather than instructing Otto to try each test separately.

Figure Four shows how this was carried out also using the Otto.exe program;

* vii). 0)Start,

1) Stand up,

2) Raise arms,

3) Take a step,

4) Take a step,

5) Test Touching Door: Yes: Open, No; Add one to memory,

6) Test Touching Door: Yes: Turn, No; Add one to memory,

7) Test Touching Anything: Yes: Repeat turn 2 times, No: Take a step,

8) Test Touching Anything: Yes: Repeat turn 3 times, No: Take step,

9) Test Touching Anything: Yes: Go to Line 6, No: Go to Line 7.

* viii): 0) Start,

1) Stand up,

2) Raise arms,

3) Take a step,

4) Take a step,

5) Test Touching Door: Yes: Open, No: Add one to memory,

6) Test Touching Door: Yes: Turn, No: Add one to memory,

7) Test Touching Anything: Yes: Turn, No: Take a step,

8) Test Touching Anything: Yes: Repeat turn 2 times, No: Take a step,

9) Test Touching Anything: Yes: Repeat turn 2 times, No: Take a step.

10) Test Touching Anything: Yes: Repeat turn 3 times, No: Take a step

11) Test Touching Anything: Yes: Repeat turn 3 times, No: Take a step

12) Test Touching Anything: Yes: Turn, No: Take a step

13) Test Touching Anything: Yes: Go to Line 6, No: Take a step

**Figure 4. Trial and Error; Attempted Layering of Commands.**

Since Otto takes thirty-eight minutes to complete each set of commands this took time to oversee. The incomplete command of Figure Five was inputted near the end of the lab after a plan was devised in which the team would ‘interchange the commands’ and try multiple sets of procedures, for example, using one of the tests to turn and repeating it three times by consistently inputting ‘Go to’ until Otto reaches his destination.

* ix.) 0)Start

1) Stand up

2) Raise arms

3) Take a step

4) Take a step

5) Test; Touching Door: Yes: Open, No: Add one to memory

6) Test; Touching Door: Yes: Turn, No: Add one to memory

7) Test; Touching Anything: Yes: Turn, No: Take a step

…

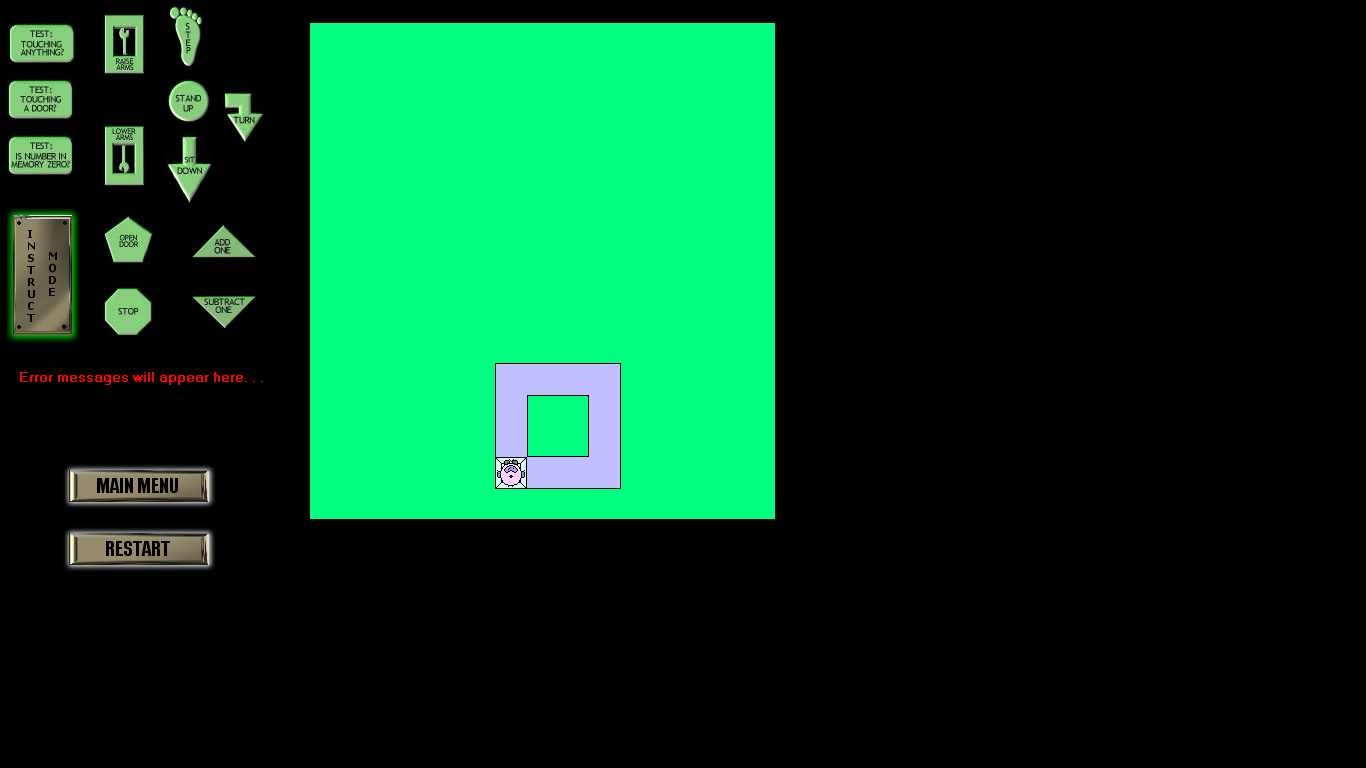
**Figure 5. Trial and Error, Incomplete Command; Example of Interchanging commands and repetition of every step.**

When we returned from Reading Week, we moved onto our programming labs and were left to our own devices to save our fictional jobs in NASA.

**Part Two: Finding Solutions**

The following three sections are the final attempts and solutions for the specified Tasks;

**Task One: Instructing Otto to walk in a Square…..**

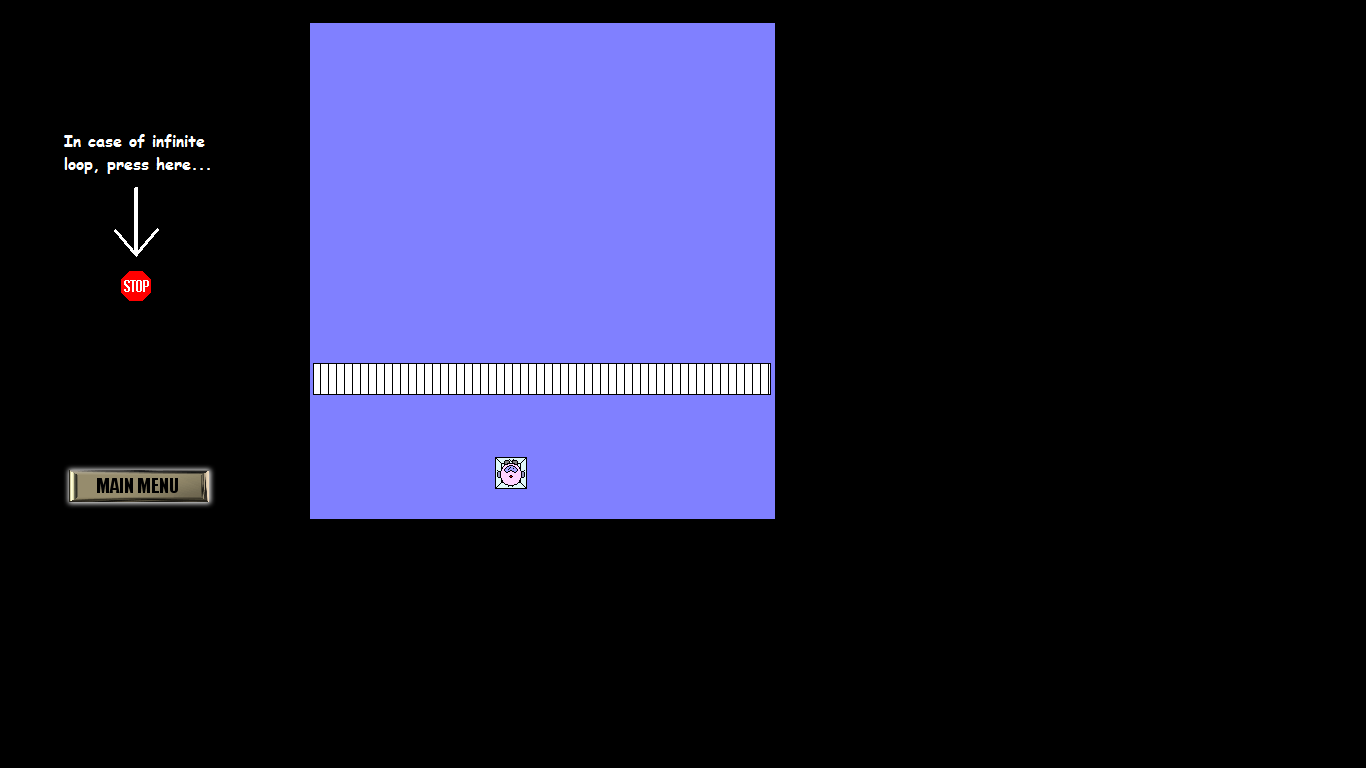


**Figure 6: Instructing Otto to Walk in a Square**

1. **Failed Commands**
2. Start,
3. Raise Arms,
4. Take a step,
5. Take a step,
6. Take a step,
7. Turn
8. Take a step,
9. Take a step,
10. Take a step,
11. Turn
12. Take a step,
13. Take a step,
14. Take a step,
15. Turn,
16. Lower arms,
17. Sit down,
18. Stop.
19. **Successful Commands:**
20. Start,
21. Raise Arms,
22. Take a step,
23. Take a step,
24. Take a step,
25. Turn
26. Take a step,
27. Take a step,
28. Take a step,
29. Turn
30. Take a step,
31. Take a step,
32. Take a step,
33. Turn,
34. Take a step,
35. Take a step,
36. Take a step,
37. Turn
38. Lower arms,
39. Sit down,
40. Stop.

**Figure 7. Instructing Otto to walk in a square commands**

**Task Two: Instructing Otto to walk to a wall…..**



**Figure 8. Instructing Otto to walk to a wall….**

1. **Successful Set of Commands:**
2. Start,
3. Stand up,
4. Raise arms,
5. Take a step,
6. Test: Touching Anything? Yes: Turn, two times, No: Go to Line 3,
7. Take a step,
8. Test: Touching Anything? Yes: Turn, two times , No: Go to Line 5,
9. Take a step,
10. Lower arms,
11. Sit down,
12. Stop.

* **Failed Commands**

1. Start,
2. Stand up,
3. Raise arms,
4. Take a step,
5. Test: Touching Anything? Yes: Repeat turn, two times, No: Add one to memory,
6. Take a step,
7. Test: Touching Anything? Yes: Repeat turn, two times, No: Subtract one from memory,
8. Go to Line 3,
9. Test: Is number in memory zero? Yes: Repeat “take a step” three times

No: Repeat “take a step” four times,

1. Repeat “turn” two times,
2. Lower arms,
3. Sit down,
4. Stop.

* **Failed Commands 2):**

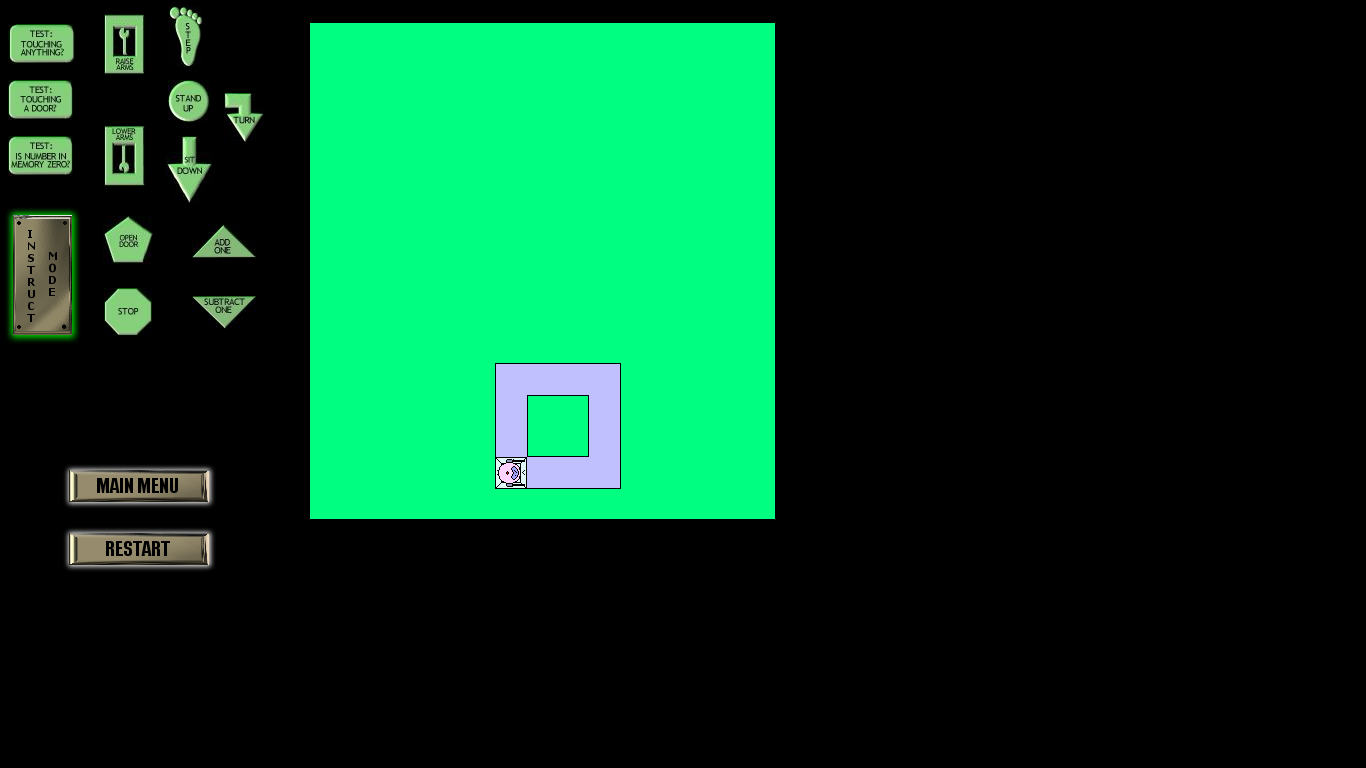
1. Start,
2. Stand up,
3. Raise arms,
4. Take a step,
5. Test: Touching Anything? Yes: Repeat turn, two times, No: Add one to memory,
6. Take a step,
7. Test: Touching Anything? Yes: Repeat turn, two times, No: Subtract one from memory,
8. Test: Touching Anything? Yes: Repeat turn, two times, No: Go to Line 5,
9. Test: Is number in memory zero? Yes: Repeat “take a step” three times

No: Repeat “take a step” four times,

1. Repeat “turn” two times,
2. Lower arms,
3. Sit down,
4. Stop.

**Figure 9. Instructing Otto to walk to a wall commands**

**Task Three: Sub-Task One:**

**Figure 10. Task Three, Sub Task One.**

**Failed Commands:**

1. Start,
2. Stand up,
3. Turn
4. Repeat ‘Take a Step’ three times.
5. Repeat ‘Turn’ three times.
6. Repeat ‘Take a step’ three times,
7. Repeat ‘Turn’ three times,
8. Repeat ‘Take a step’ three times,
9. Repeat ‘Turn’ three times,
10. Repeat ‘ Take a step’ three times.
11. Lower arms,
12. Sit down,
13. Stop.

0) Start

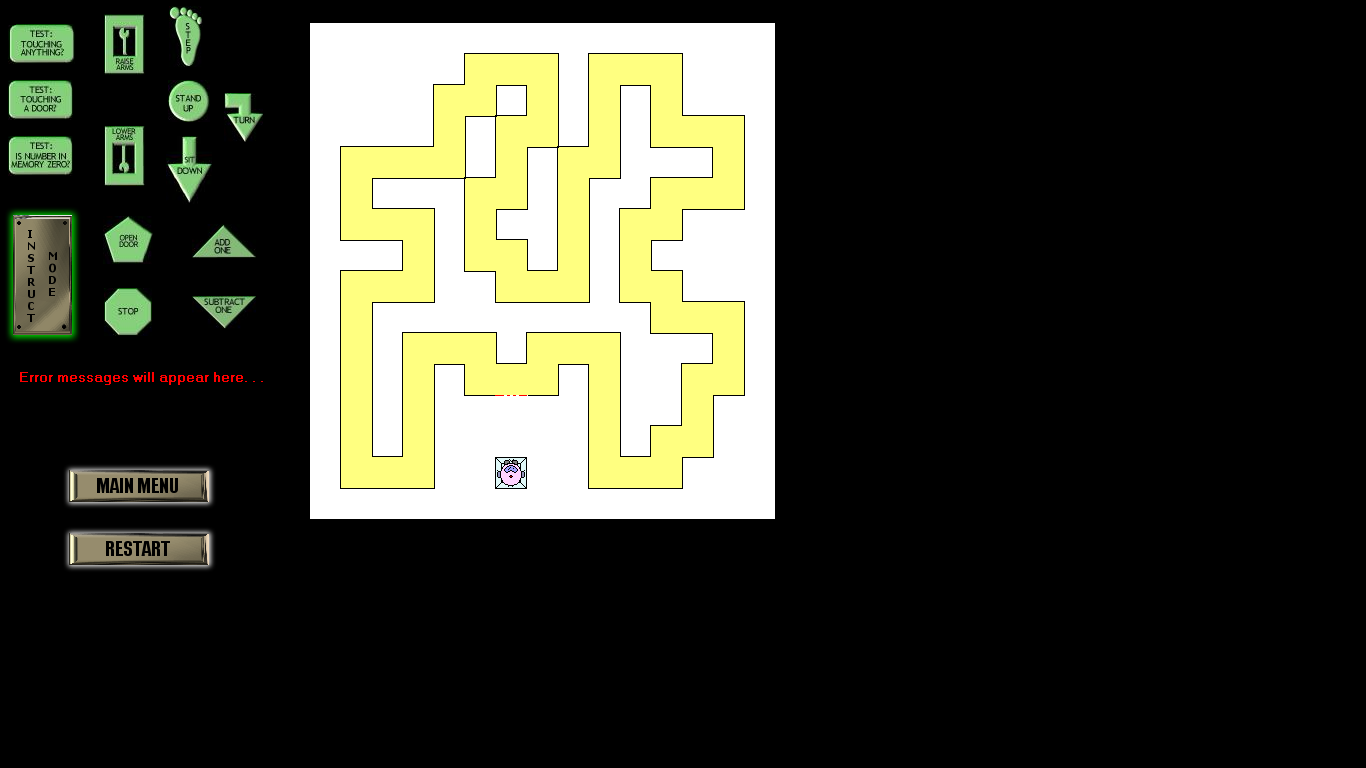
1. Stand up,
2. Raise arms,
3. Turn.
4. Repeat ‘Take a Step’ three times.
5. Repeat ‘Turn’ three times.
6. Repeat ‘Take a Step’ three times.
7. Repeat ‘Turn’ three times.
8. Repeat ‘Take a Step’ three times.
9. Repeat ‘Turn’ three times.
10. Repeat ‘Take a Step’ three times.
11. Repeat ‘Turn’ three times.
12. Lower Arms.
13. Sit down
14. Stop.

**Successful Commands:**

1. Start
2. Stand-up
3. Raise arms
4. Turn
5. Repeat ‘Take a step’ three times
6. Repeat ‘Turn’ three times
7. Repeat ‘Take a step’ three times
8. Repeat ‘Turn’ three times
9. Repeat ‘Take a step’ three times
10. Repeat ‘Turn’ three times
11. Repeat ‘Take a step’ three times
12. Repeat ‘Turn’ two times
13. Lower arms
14. Sit down
15. Stop

**Figure 11: Commands for Task Three: Sub-Task One.**

**Task Three: Sub-Task Two:**

**Figure 12: Task Three: Sub-Task Two(Screenshot: Example with one door).**

0) Start,

1) Stand up,

2) Raise Arms,

3) Take a step,

4) Test: Touching Door? Yes: Open Door, No: Go to Line 3,

5) Take a step,

6) Test: Touching Anything? Yes: Turn, No: Go to Line 5,

7) Test: Touching Anything? Yes: Turn, No: Go to Line 5,

8) Turn,

9) Test: Touching a Door? Yes: Open Door, No: Go to Line 7,

10) Repeat ‘Take a Step’ ‘3’ times,

11) Turn,

12) Repeat ‘Take a Step’ ‘3’ times,

13) Turn,

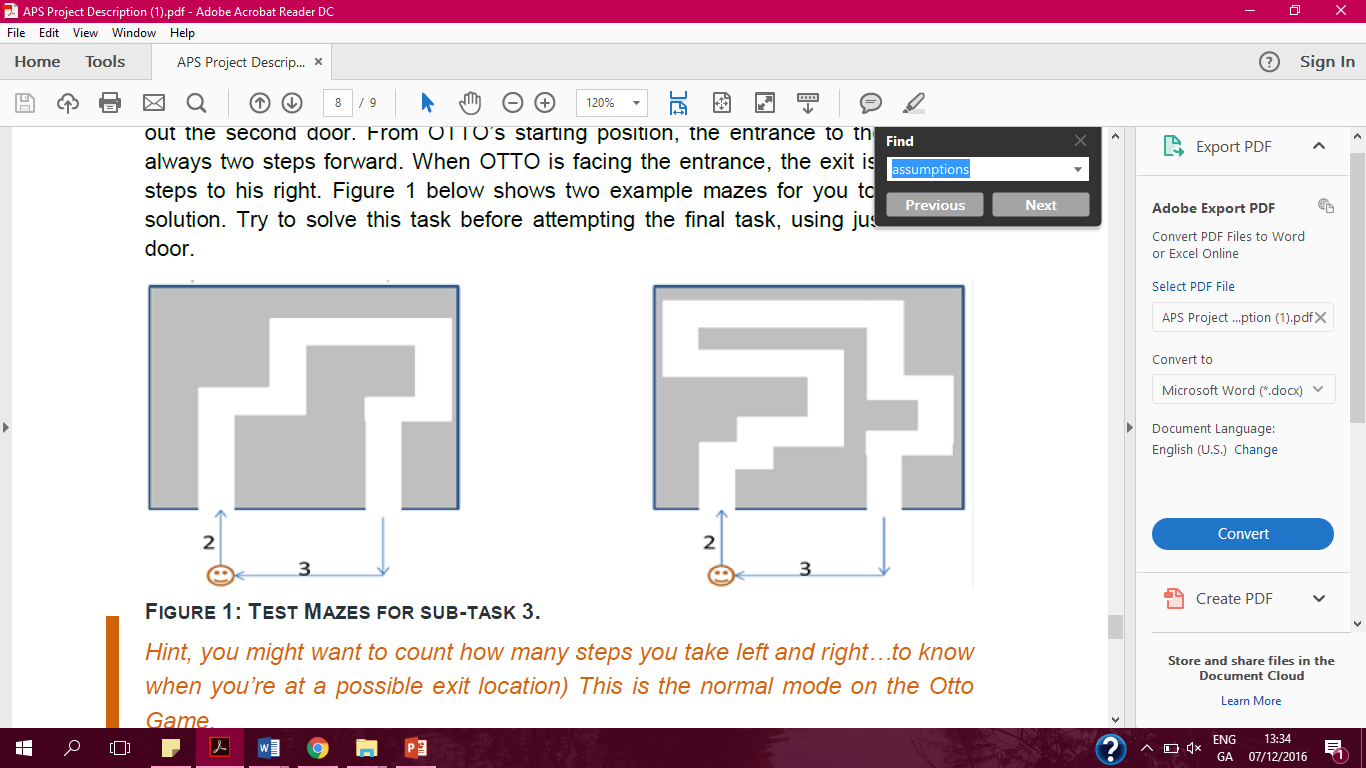
14) Lower arms,

15) Sit down,

16) Stop.

**Figure 13. Task Three: Sub-Task 2 Commands.**

**Task Three, Sub-Task Three:**



**Figure 14. Test Mazes for Sub-Task Three from The Project Description.**

1. **Start:**

1) Stand up,

2) Raise arms,

3) Take a step,

4) Take a step,

5) Test: Touching a door? Yes: Open, No: Take a step,

6) Take a step,

7) Repeat: Turn: 3 times,

8) Take a step,

**Body:**

9) Test: Touching a Door? Yes: Go to Line 11, No: Turn,

10) Test: Touching Anything? Yes: Repeat: Turn 2 times, No: Take a step,

11) Test: Touching a Door? Yes: Open, No: Go to Line 9,

**End:**

12) Repeat: Take a step: 3 times,

13) Turn,

14) Turn,

15) Lower arms,

16) Sit down,

17) Stop.

**Figure 15. Task Three, Sub-Task 3 Commands.**

**Our Approach to the Tasks**

The logic used to solve ‘Task Two: Instructing Otto to walk to a wall…’ can be seen in the first part of this report.

At first, we thought solving Sub-Task One was just going to be as simple as solving the first task. Under the limitations, Otto cannot turn left, so we must order to make three right turns to successfully manoeuvre around the square. When we used the same commands as Task One with this additional repeated commands, the program failed. We then added the command to make Otto turn twice at the end to position him correctly on his charging pad before placing him back onto it. In Sub-Task Two that Otto will enter and leave the maze at the same point was taken into account. During the second lab. the team decided to use the “Go to” command to implement a loop that would allow the robot to open any door that would stand in his way instead of inserting separate commands that would get him to take two steps and then open that specific door. This loop ensured that the whole set of commands didn’t fail as OTTO would continue looking for a door until he found one.

This sounds easy but some problems were encountered in the process. For example, the robot continuously ended up in loops and wouldn’t progress past a certain corridor. If this was a real situation the ‘Touching Anything? Yes: Turn, No: Take a step’ Command would eventually cause Otto to go back on himself since he can only turn in one direction. The ‘Take a Step’ Command was applied, before telling the robot to take a step if it was touching anything. This application permitted the creation of another loop. This is where the second problem arose, OTTO became stuck in a certain part of the maze.

This is where duplication or the layering of commands as seen earlier in this report came into play. The “Touching Anything?” instruction was used that is the same as the one above it in conjunction with “Turn” and “Test: Touching Door?” commands. The addition of the “Test: Touching Door? Yes: Open, No: Go to Line 7” command leads to repetition of two loops until OTTO found his way out of the maze. This test also solves the problem of Otto getting stuck with it allowing him to take three turns if necessary and look for the door by looping back to Line 5. The “Turn” command is vital in preventing OTTO from exiting the maze the same way it entered. The duplication of commands ensures that a whole set doesn’t fail on just one incorrect instruction causing the loss of billions of imaginary dollars for the agency.

Returning OTTO to his charging pad in this maze was difficult to achieve as with something this complicated there is a wide range of instructions that could have been used successfully. The commands shown in Figure Nine are the ones that worked for the team.

The maze from the final sub-task involved a lot of creativity and patience. To order OTTO to successfully navigate through this maze, return to the starting position and come to a complete stop it required a lot of trial and error to find the correct commands and experimentation to see if they would work. The process to the above solution was rather long but fruitful. The possible commands for the other mazes in which the “Body” as seen in Figure Eight is the only thing that stays the same for the rest. The “Start” of each set of instructions changes because of Otto being an unknown number of steps away from an obstacle. The “End” also does not stay the same because the starting and ending point will vary with the complexity of each maze.

The list of commands shown above were successful but are only compatible for the above maze. The conclusion that a lot more trial and error would be needed to make a ‘one size fits all’ set of instructions. Perhaps, all three of the maze’s commands could be put together and one could work from there if they were to undertake such a task.

**Conclusion:**

This report is the culmination of six weeks of lots of hard work and effort put in by every member of the team. The labs that covered Otto, the Robot, were useful in the organisation and discussion about the project and helped us to focus our minds on the issues at hand. When we moved on to APS Programming the commands and the mazes were never far from our minds as we sought to complete the project.

OTTO, the Robot, also reminded us of what we learned in the first few weeks of this module: the importance of writing things down, testing them out and how two or in this case four heads are better than one in solving problems. The feeling when each task was finally solved was one to remember. It proved that using one’s head is above and beyond just searching the Internet when trying to come to a resolution to a problem. It truly felt as if we had just saved years of research, invested money and our reputations at NASA.

**Index:**

**Figure 1. Sample Brainstorm from Lab 1.**

**Figure 2. Additions to the sample brainstorm.**

**Figure 3. Trial and Error; Discarded Commands 1-6.**

**Figure 4. Trial and Error; Attempted Layering of Commands.**

**Figure 5. Trial and Error, Incomplete Command; Example of Interchanging commands and repetition of every step.**

**Figure 6. Instructing Otto to walk in a square……**

**Figure 7. Instructing Otto to walk in a square Commands.**

**Figure 8. Instructing Otto to walk to a wall….**

**Figure 9. Instructing Otto to walk to a wall Commands**

**Figure 10. Task Three Sub-Task One.**

**Figure 11. Commands for Task Three Sub-Task One**

**Figure 12. Task Three Sub-Task Two (Screenshot: Example with one door).**

**Figure 13. Task Three Sub-Task Two Commands.**

**Figure 14. Test Mazes for Sub-Task Three from The Project Description.**

**Figure 15. Task Three: Sub-Task Three.**