**Comp H1034-**

**A.P.S Otto the Robot Project Report.**

* Timothy McCann

(B00095944).

* Raigridas Bartkus (B00094651).
* Benjamin Scallan

(B00097614).

* Leah Brennan

(B00098851).

[Group Seven].

**Table of Contents**

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

Part One: Form, storm, norm and perform……………………………………………………4-10.

Part Two: To infinity and beyond the schematics…………………………………………...11-19.

1. The *Easy* Maze, Solved by Ben……………………………………………..12-14.
2. The *Medium* Maze, Solved by Tim…………………………………………15-17.
3. The *Hard* Maze, Solved by Raigridas……………………………...………..18-19.

Conclusion………………………………………………………………………………………20.

Index……………………………………………………………………………………………21.

**Introduction:**

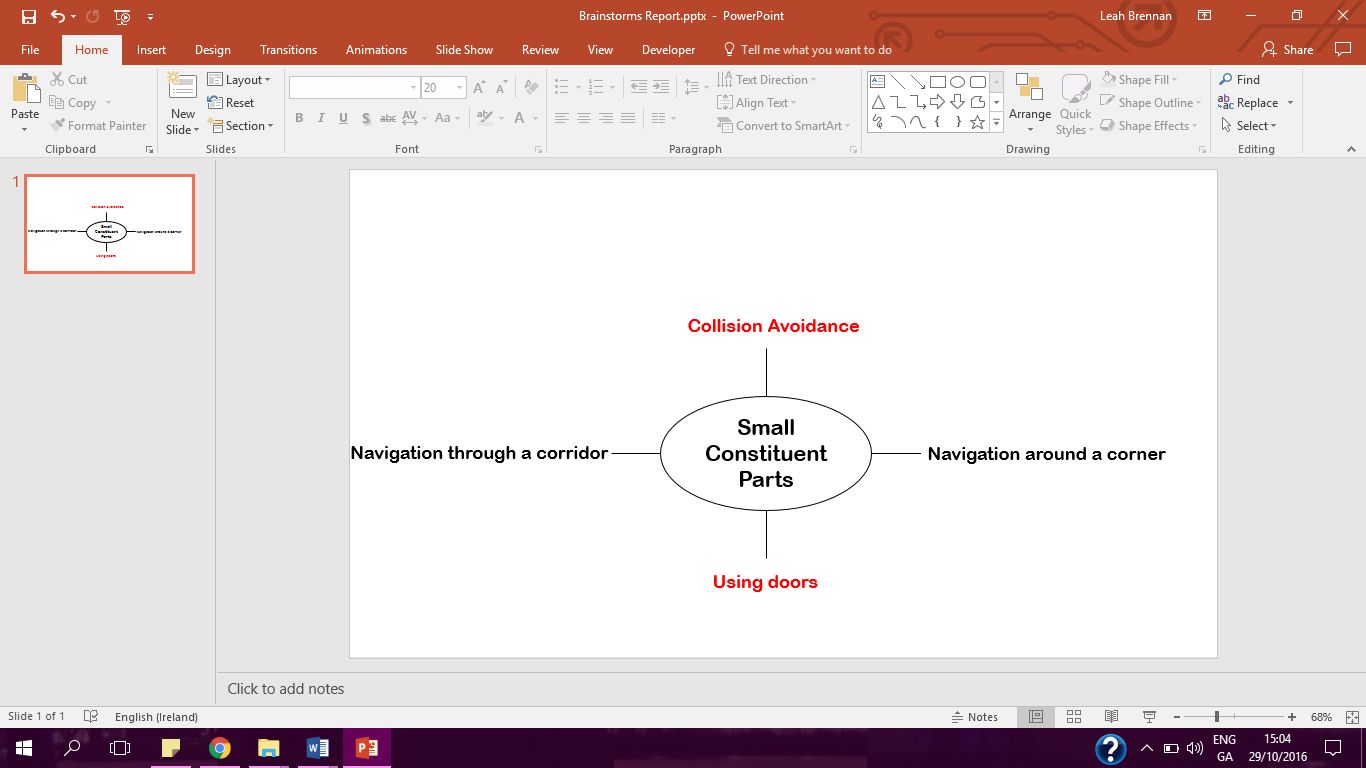
This was the first group project that the group were assigned. We were relatively strangers before this started so we didn’t know each other’s individual strengths. However, over the period of the project everyone’s separate talents were brought to bear enthusiastically. The completion of Otto the Robot has spanned over a month and a half 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 quicker 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: Form Storm Norm and Perform.**

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, Ben, Tim and Raigridas took the project back to earth. Ben 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. Tim and Raigridas added 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.

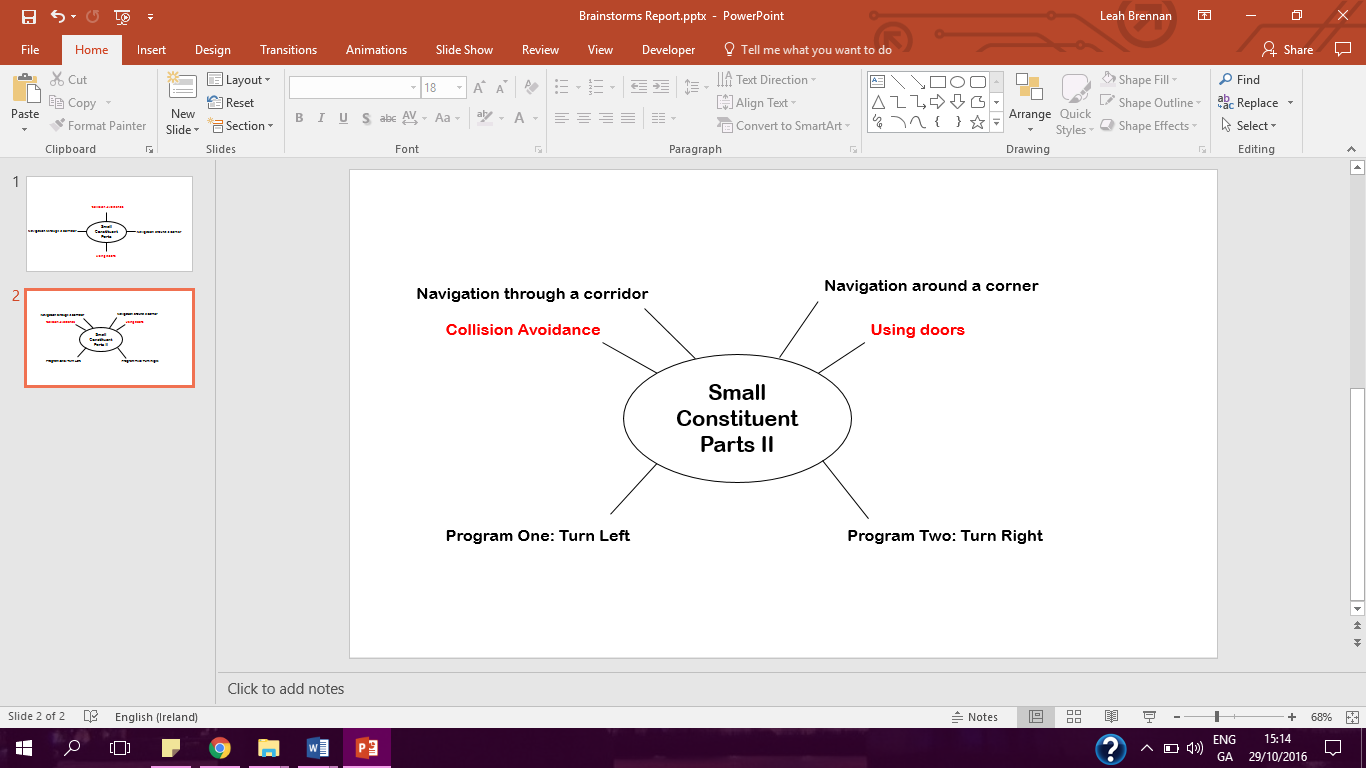


**Figure 1. Sample Brainstorm from Lab One.**

With Gerard’s team’s contribution and Ben, Tim and Raigridas’ 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. Following the group discussion on the report each team member’s role was drafted; Ben agreed that he would complete the task that the team labelled; The *Easy* Maze, Tim decided to take The *Medium* Maze and Raigridas would tackle The *Hard* Maze. 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

Ben then questioned Otto’s motivations for going into a maze and the robot’s definition of a door. He then brought it back to his idea from the first lab that the maze should be broken down into its ‘small constituent parts’ this time to 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 go through any maze.



**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). 1) Start,

2) Stand up

3) Raise arms,

4) Take a step,

5) Take a step,

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

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

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

* ii). 1) Start

2) Stand up,

3) Raise arms,

4) Take a step

5) Take a step

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

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

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

* iii). 1) Start

2) Stand up,

3) Raise arms,

4) Take a step

5) Take a step

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

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

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

* iv). 1) Start

2) Stand up

3) Raise arms,

4) Take a step,

5) Take a step

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

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

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

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

* v). 1) Start

2) Stand up

3) Raise arms

4) Take a step

5) Take a step

6) Test; Touching Door; Yes; Open, 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: Take a step

* vi).1) Start

2) Stand up

3) Raise arms

4) Take a step

5) Take a step

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

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

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

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

10) 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 a suggestion was put forward to try ‘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). 1)Start

2) Stand up

3) Raise arms

4) Take a step

5) Take a step

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

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

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

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

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

* Next): 1) Start

2) Stand up

3) Raise arms

4) Take a step

5) Take a step

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

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

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

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

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

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

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

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

14) 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.

* Next). 1) Start

2) Stand up

3) Raise arms

4) Take a step

5) Take a step

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

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

8) 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: To infinity and beyond the schematics.**

The following three sections are the attempts and the solutions for the problems that the team labelled; The *Easy* Maze: Solved by Ben, The *Medium* Maze: Solved by Tim, The *Hard* Maze: Solved by Raigridas:

**The *Easy* Maze, Solved by Ben:**

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 4.
7. Take a step.
8. Test: Touching Anything? Yes: Turn, two times, No: Go to Line 6.
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 4.
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 6. The *Easy* Maze, solved by Ben.**

The logic and plan used in Ben’s maze is described earlier in the report when the team’s discussion from the First and Second Lab that covered the project is written.

**The *Medium* Maze, Solved by Tim**

1. Stand up

2. Raise Arm

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. Take a step

11. Take a step

12. Take a step

13. Turn

14. Take a step

15. Take a step

16. Take a step

17. Turn

18. Lower arms

19. Sit down

20. Stop

**Figure 7. The *Medium* Maze, Solved by Tim**

To solve the Medium Maze, Tim focused on how OTTO’s recharge pad was two steps away from the door. 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 Tim encountered some problems 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. Tim then applied the method that the team agreed on. He applied the ‘Take a Step’ Command before telling the robot to take a step if it was touching anything. This application allowed him to create another loop. This is where Tim met with his second problem, OTTO would become stuck in a certain part of the maze.

Tim decided to attempt to duplicate or layer the commands as seen earlier in this report. He used the “Touching Anything?” instruction 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 Tim and the team.

**The *Hard* Maze: Solved by Raigridas**

**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 8. The *Hard* Maze, Solved by Raigridas.**

This maze involved a lot of creativity and patience for Raigridas. For him to order OTTO to successfully navigate through this maze, return to the starting position and come to a complete halt 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. He looked at 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 Raigridas’ maze. He arrived at the conclusion that a lot more trial and error would be needed to make a ‘one size fits all’ set of instructions. Raigridas suggested that 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. The *Easy* Maze, Solved by Ben.**

**Figure 7. The *Medium* Maze, Solved by Tim.**

**Figure 8. The *Hard* Maze, Solved by Raigridas.**