AI1: Sokoban Project Group 22

Michael K. Schmidt, micsc12@student.sdu.dk
Stefan J. M-Møller, stmae09@student.sdu.dk

Course: RMAI1 - Artifial Intelligence
T: John Hallam, John@mmmi.sdu.dk
TA: Danish Shaikh, danish@mmmi.sdu.dk

February 6, 2017

1 Preface

This report describes the work done for the Sokoban Project during the fall semester 2016 in the RMAI1 course. The report has been split up into two parts, as per suggestion of our teacher, John Hallam. The first part being Robot which goes into detail with the evolution, behaviours and various tests. The second part being the solver/planner which goes into detail with the implementation, choosing a heuristic and a walk-through of the algorithm.

Contents

1 Preface **1**

2 3.1 Global Navigation Satellite Systems **2**

3 3.1 Global Navigation Satellite Systems **2**

4 Robot **3**

5 Conclusion **4**

References **4**

2 3.1 Global Navigation Satellite Systems

3 3.2 GPS architecture

4 3.3 GNSS error sources

5 3.4 Dilution of Precision (DOP)

6 3.5 GNSS accuracy

7 RTK-GNSS

8 Robot

9 Conclusion

In conclusion the robot performed sub-par. The design became heaps better with the introduction of the Multi-bot design and was able to mount sensors etc. virtually everywhere. The robot went through some tests at every evolutionary step, which helped defined the shape and functionality of the robot. Gearing of the robot was made too aggressively slow and as a result we had a very robot. the gearing should instead be made opposite so we can increase the speed and adjust the behaviours accordingly. Further tests have been devised for evaluation the new gearing design.

The planner implementation was found to be performing relatively well, being able to handle complex graphs (167 million nodes was explored for the 2016 competition map).

It was able to return a greedy solution for most maps very rapidly, because of the optimal datastructures chosen for lookup of identical states and choosing what state to explore next.

Further work should go into lowering the memory requirements of each state to make it possible to solve even more complex maps. Also, it should be investigated if some parts of the tree can be removed from memory, after it has been explored.

References

[1]

[2] Mark A. Weiss *Data Structures and Algorithm Analysis in C++*, Pearson International Edition 4rd edition, 2014.