



UNIVERSITY OF
LINCOLN

School of Computer Science
Assessment Package Briefing Document

**Title: CMP2067M Advanced Software
Development Assignment 2**

Indicative Weighting: 50%

Learning Outcomes:

On successful completion of this assessment package a student will have demonstrated competence in the following areas:

- [LO3] Use advanced OO principles and programming techniques in software development;
- [LO4] Apply advanced logical and mathematical techniques in software development and programming.

Path-finding Application

Path-finding is used in many different application areas, for example in map directions, routing an emergency vehicle to a destination quickly, games and others.

The objective of this assignment is to:

- Design and implement a path-finding A* algorithm;
- Research and implement a non-trivial extension of your algorithm;
- Write a technical report that summarises your approach and the Object Oriented principles that you have employed.

More specifically, you will have to address the following points:

- Your implementation should enable an “agent” to find an optimal path from its starting position to any goal location specified for a given **REAL** environment. You should use University environments – for example a path from a start point on the 3rd floor of the MHT building to an end point on the 3rd floor of the MHT building.
- The form of the environment will be represented by a grid of cells with obstacles/walls. Inputs to your code should include (but not limited to)
 1. Start and goal positions
 2. Map file of the workspace and dimensions of the workspace (i.e. number of cells in the x and y dimensions).
 3. The map file should be a .txt file in a similar format as the one expected by the `readTXT` function provided in Assignment 1. Walls should be represented by 0s and walkable cells by 1s.
 4. You should create your own workspaces and test your algorithm on them. An example of an environment is shown in Fig. 1.

The output of your code should be the optimal path. You should write a main program which provides the necessary inputs to your algorithm and then calls the algorithm to find the optimal path.

In your submission, you should address the following:

- A* implementation: Your design and implementation should employ Object Oriented principles and data structures (e.g. lists, cells/nodes, grids/graphs, arrays). All code should be written in C++. A thorough analysis of your implementation should be described in your report.
- A* extensions: You should research and implement a non-trivial extension to the basic A* algorithm. Possible extensions that you may want to consider are (but not limited to, in this section it is highly recommended that you implement anything that you really like!)
 1. More than one (moving) agents – Collision avoidance
 2. Graphical user interface (inputs, illustrating the environment and the evolution of the search etc.)
 3. Theoretical extension of A* (for example D*, D* Lite or hierarchical A*)

A thorough analysis of the extensions which you implement should be described in your report.

Your report must be **NO MORE** than 8 pages long and should contain the following sections:-

- Introduction: Your understanding of A* algorithm.
- Methods: Design (Object Oriented Principles and Data Structures that you employed) and Pseudo-Code of your solution.
- Results: Evidence of your program running with the expected output.
- Extensions: A thorough analysis and discussion on non-trivial extensions implemented in the solution.
- A list of references used.

Demonstration of Submissions

There will be a demonstration activity associated with this assignment as well as the submission detailed below, the exact time/place of which will be communicated to you by the teaching team. Not attending the allocated session, will incur a penalty.

Submission Guidelines

Please zip up your project files (which should include an executable file, and your source files, along with any other accompanying files) as a compressed zip file and should be submitted through Blackboard

The written report should be submitted separately on Blackboard to the relevant Turnitin submission sit

This module is graded using a criterion reference grid. You should be clear in your understanding of the grading principles; if you are not, please seek the advice of the module co-ordinator.

Hand In Instructions

See hand in schedule.

DO NOT include this briefing document with your submission.

