COMP3702/COMP7702 Artificial Intelligence (Semester 2, 2020) Assignment 1: Search in LASERTANK— **Report Template**

Name: Anindya Sarup

Student ID: 44243726

Student email: a.sarup@uqconnect.edu.au

Note: Please enter the name, student ID number and student email to reflect your identity and **do not modify the design or the layout in the assignment template**, including changing the paging.

Question 1 (Complete your full answer to Question 1 on the remainder page 1)

The dimensions of complexity in LASERTANK have been stated below, along with the the reasoning of their selection:

- **Modularity:** The modularity for LASERTANK is **flat**. There is no organizational structure, that is, the agent does not reason on multiple levels of abstraction and the environment is discrete.

- **Planning Horizon:** The planning horizon for LASERTANK is an **indefinite stage problem** because while the agent looks some finite steps ahead, but it does not know the total amount of steps it would take for the agent to get to the goal.

- **Uncertainty:** The environment is partially observable to the agent and the effect of uncertainty is deterministic because the state resulting is determined from the action of the agent on the state. Thus the sensing uncertainty dimension is **partially observable** and the effect of uncertainty is **deterministic**.

- **Goals versus complex preferences:** The agent is goal driven with no trade-offs involved. Thus its preference is an **achievement goal**.

- **Single or multiple agents:** It is **single agent** driven because there is only a single agent performing an action in the environment.

- **Learning from experience:** There is no learning from past data or experience involved, and the **knowledge is given**.

- **Perfect rationality versus bounded rationality:** The agent reasons for the best action without taking into account its limited computational resources and thus has **perfect rationality**.

- **Succinctness and Expressiveness:** The environment has multiple features with each feature containing value assigned by the state.

Reference: `https://www.cs.ubc.ca/~poole/papers/dimensions2006.pdf`

Question 2 (Complete your full answer to Question 2 on page 2)

- **Action Space:** The set {*MOVE_FORWARD, MOVE_LEFT, MOVE_RIGHT, SHOOT_LASER*}. The set mentioned is the set of all possible actions that the agent can perform.

- **Percept Space = Perception Function:** The tiles adjacent to the agent are the only agent in LASERTANK can perceive. Thus, the percept space for the agent is the set of all adjacent tales for the agent at a particular position (its neighbourhood).

- **State Space:** The state space of this environment are all the tiles present in the LASERTANK map.

- **Transition Function:** The resultant change of the configuration of the world after the agent performs actions on it are described here:

  - **MOVE_FORWARD:** When the agent performs the action MOVE_FORWARD, the following takes place:
    * *If there is nothing:* If there is nothing on the tile in the heading direction of the player, the player moves one tile forward with no consequences.
    * *If there is a anti-tank:* If there is a anti-tank in the line of sight of the player's heading direction, the consequence is game over.
    * *If there is a wall or mirror:* In the case of a wall or a mirror in the heading direction of the player, there is no change to the player's position.
    * *If there is ice:* In the case of ice, the player glides forward on the ice until there is no more ice or until a COLLISION takes place.
  - **MOVE_LEFT or MOVE_RIGHT:** This command only results in changing the heading direction of the player, thus leading to the change in percept space of the player/tank.
  - **SHOOT_LASER:** This action causes a change in the change of the state space, as characters on the game may lose their position

- **Utility Function:** If the player/tank reaches the target/flag, the value for the utility function is 1 otherwise it's 1. This is due the fact that the agent has no objective apart from reaching the target/flag.

Question 3 (Complete your full answer to Question 3 on page 3)

Performance of **Uniform Cost Search** and Performance of **A* Search** on the file **t3_the_river.txt**:

Question	Uniform-Cost-Search	A* Search
(a) The number of nodes generated	57241	29029
(b) The number of nodes on the fringe when the search terminates	390	397
(c) The number of nodes on the explored list (if there is one) when the search terminates	14701	7655
(d) The run time of the algorithm (in seconds)	3.5417 seconds	1.7378 seconds

(e) The test-case *'t3_the_river.txt'* has been chosen due to the intricacies involved in finding an optimal solution for this test-case.

It can be clearly seen that the total number of nodes generated during A* Search are almost exactly half of the total number of nodes generated during UCS algorithm-based implementation. This clearly affects the total run-time of the algorithms, where A* star takes half the time to find the most optimal solution when compared to UCS.

UCS visits/explores twice the number of nodes when compared to A* Search because it gives equal importance (weight) to all items in the action space set, while A* is biased due to its heuristic function.

All these factors directly affect the total run-time of the two algorithms, thus one can see the discrepency in performing an uninformed search (UCS) and an informed search (A* algorithm).

Question 4 (Complete your full answer to Question 4 on page 4 and 5)

Due to the large dimensions of the different test-cases provided for this assignment, **'Manhattan Distance'** was chosen as the heuristic function.
Due to the nature of how the game works, a different heuristic function was not chosen because the special tiles - Ice, bricks, mirrors and teleporter tiles were already well-implemented internally. Also due to the nature of my step function that uses the 'apply_move(move)' function of LaserTankMap class, I decided not to create different heuristic functions for the game.

In hindsight, maybe a different approach to the entirety of the 'solver.py', step-function and heuristics could've resulted in a better inclusion of the special tiles.

. . .