Sumanth Donthula(A20519856)

**CS 480 Spring 2023 Written Assignment #01**

Due: **Thursday, February 8, 2023, 11:53 PM CST**

Points: **20**

**Instructions:**

1. Use this document template to report your answers. Name the complete document as follows:

LastName\_FirstName\_CS480\_Written01.doc

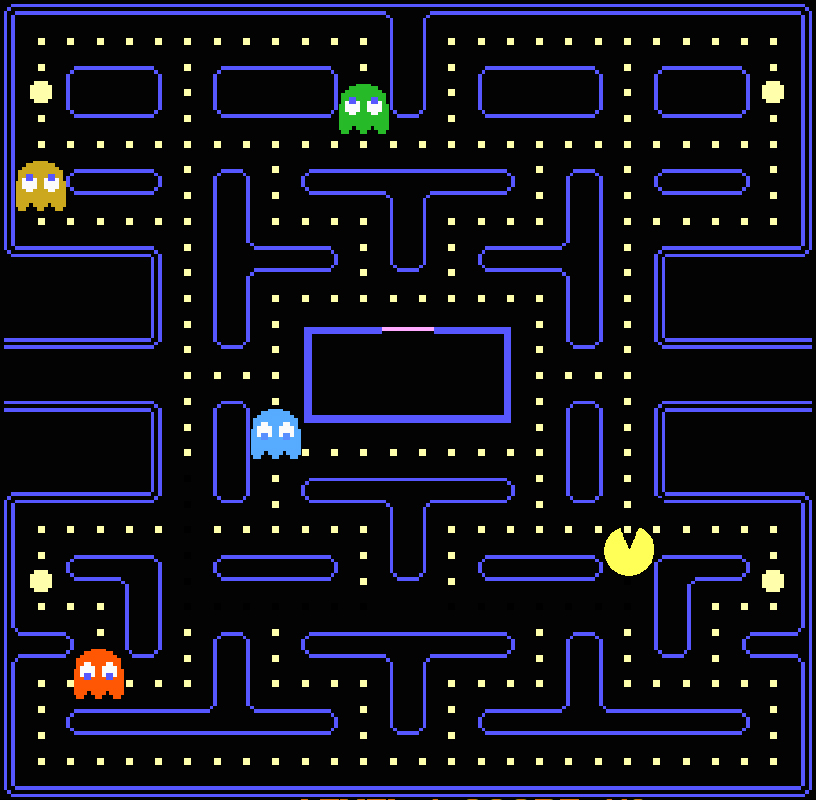
1. Submit the final document to Blackboard Assignments section before the due date. No late submissions will be accepted.

**Objectives:**

1. (8 points) Analyze an agent-environment system and apply the PEAS agent description.
2. (12 points) Demonstrate your understanding of a simple informed search algorithm.

**Problem 1:**

Consider the game of Pac-Man (fig. 1). More on the game can be found here: <https://en.wikipedia.org/wiki/Pac-Man#Gameplay>



*Figure 1: Pac-Man game screenshot .*

Your task is to:

* decide what is the agent and what is the environment in this system **[1 pt]**:

|  |  |  |
| --- | --- | --- |
| **Agent** | **Environment** | **Explanation** |
| Pac-Man and Ghosts | Maze and Walls in Maze, and Dots(Food) in the Maze | In this case Pac-Man and ghosts are agents acting upon the environment which is Maze which contains Dots(Food). |

* analyze the system and apply the the PEAS (Performance measure, Environment, Actuators, Sensors) description **[3 pts]**:

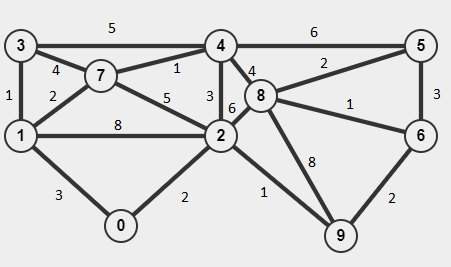
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Performance measure** | **Environment** | **Actuators** | **Sensors** | **Explanation** |
| Finding the optimal path to consume more dots and avoid ghosts. | Maze and Walls in Maze and Dots(Food) in the Maze | Keyboards strokes to say whether agent should move North(Up), West(Left), East(Right) and South(Down) | Observations whether a path in a direction of the maze is closed or open, whether Ghost is present, whether Dots (Food) is present. | * The goal of a Pac-Man is to eat all Dots without been eaten by ghosts and find the optimal path in the maze. * So the performance measure will be to eat maximum dots. * The environment will be the maze walls, dots and ghosts. * The keyboard inputs the agent to act upon in environment. * The Observations like whether a path is available or whether ghost is far are sensors in the case. |

* Specify the properties of this environment. Justify your decisions **[4 pts]**:

|  |  |  |
| --- | --- | --- |
| **Property** | **Your choice** | **Explanation** |
| Fully observable? | Fully Observable | As we can see all the boundaries of the Game, it is Fully Observable. |
| Multiagent? | Multi Agent | We have Pac-Man and Ghosts as agents. So, it’s a multi agent scenario. |
| Deterministic? | Deterministic | We can move the Pac-Man by current state and the ghosts in the game. |
| Episodic? | Sequential | The current decision depends on previous decision as for example let’s assume a Pac-Man is eaten by Ghost, it don’t have possible next step. So, current decision is dependent on previous one. |
| Dynamic? | Static | The Environment is static as the walls of the Maze are not moving. |
| Discrete? | Discrete | The agent’s steps are discrete values on the maze. |
| Known to Agent? | Known | The Pac-Man knows the environment as it/player knows the rules and plays, it will move in the Maze accordingly. |

**Problem 2:**

Consider the **undirected** graph presented below (fig. 2). Each node represents a single state (you can assume that each state represents a city on a map). If two states are neighbors, there is an edge between them.



*Figure 2: An undirected graph.*

Assume that edge weights represent **driving distances between cities/states in miles**.

Your task is to utilize the **Hill Climbing algorithm to find a shortest (minimum cost) path** between two states provided data. Here are the steps:

* assume that **repeated states are NOT allowed**,
* select two states / cities (initial and goal states) at random under the condition that there is at least two (2) states between your initial and state goals (that would correspond to **at least** three (3) actions),
* apply the Hill Climbing algorithm and show all steps / actions in Table A below,
* provide a search tree diagram illustrating the path chosen along with evaluation function values and all alternatives (you can paste in a scan or a photo of a hand-drawn diagram or use some software to create it).

**NOTE: The algorithm may get “stuck” and not reach the goal state.**

**Initial State : 1**

**Goal State : 6**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TABLE A: Algorithm steps / actions [6 pts]** | | | | |
| Current state | Available actions and their costs | Selected action | Resulting state | Explanation / comments |
| 1 | States: 7,2,3,0  Cost : 2,8,1,3 | 1----->3 | 3 | We chose State 3 as its cost is minimum i.e, 1 |
| 3 | States: 4,7  Cost : 5,4 | 3----->7 | 7 | We chose State 7 as its cost is minimum i.e, 4 |
| 7 | States: 4,2  Cost : 1,5 | 7----->4 | 4 | We chose State 4 as its cost is minimum i.e, 1 |
| 4 | States: 8,5,2  Cost : 4,6,3 | 4----->2 | 2 | We chose State 2 as its cost is minimum i.e, 3 |
| 2 | States: 8,9  Cost : 6,1 | 2----->9 | 9 | We chose State 9 as its cost is minimum i.e, 1 |
| 9 | States: 8,6  Cost : 8,2 | 9----->6 | 6 | We chose State 6 as its cost is minimum i.e, 2 |
|  |  |  |  |  |
|  |  |  | **Total Cost** | **12** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |
| --- |
| **Tree search diagram [6 pts]** |
| C:\Users\SRINU\Downloads\New Document(8)_1.jpg |

Did the Hill Climbing algorithm pick the best (lowest total cost in miles) path?

|  |
| --- |
| **Your answer:** |
| I think Hill Climbing Algorithm will always not pick the best path which minimizes the cost. It always can be blinded by the local minima as its scope while taking a next step will be obtaining the path which minimizes the cost from current state to the next state. For instance, in the example I have used to determine path from State 1 to State 6. The alternative path could have been 1🡪0🡪2🡪9🡪6 whose total cost is **8**. The cost of path obtained by Hill Climbing algorithm is **12** which is not optimal. It took a step which takes a path 1🡪3 at starting state instead of 1🡪0 which misinterpreted the path as it’s cost is minimum. Moreover, in this case we have reached our goal but in some cases the algorithm fails to find a path between initial and final states. |