Vietnam National University
Bach Khoa University
Faculty of Information Technology

Assignment

Course: Introduction to Artificial Intelligence

Duration: 03 weeks

I. Formation

- The midterm project is conducted in groups of 04 – 05 students.
- Student groups conduct required tasks and submit the project following instructions.

II. Tasks

a) Task 1 (4.0 points): 8-Puzzle

In the game 8-puzzle, you have to move tiles, given an initial state, to obtain one goal state. Note that

- The initial state can be arbitrary permutation of 8 tiles and the blank cell.
- There are two goal states illustrated in the table below.

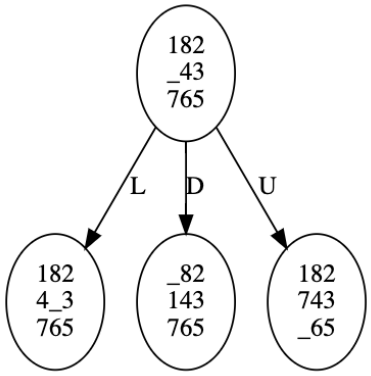
Initial state	*Goal state*

1 2 7
4 6 8
  5 3
 	
…
	1 2 3
	4 5 6
	7 8
	or
	  1 2
	3 4 5
	6 7 8

Students conduct the following requirements:

- Formulate the given problem in form of a state space, determine details of related components.
- Implement BFS and A* algorithms to solve the game.
- Implement a complete program to execute designated algorithms, in which

- Input: an initial state provided by the user; name of the expected algorithm
- Output: list of actions (Left, Right, Up, Down); total cost
- Visualization: students visualize steps of the game on the console screen or use the **graphviz** library to draw the path from the initial state to the goal in Google Colab notebook (ignore nodes outside of the path).



- For A* algorithm, students propose at least one heuristic function and determine its admissibility and consistency.
- Implement a program, in which
  - Generate randomly 1000 initial states
  - For every initial state, execute BFS and A* algorithms, compute the path cost for each algorithm
  - Compute the average cost for each algorithm
  - Draw a bar chart, using Matplotlib, to illustrate average path costs above.
- Students organize the program regarding to the OOP model, ensure source code is compact and rational.

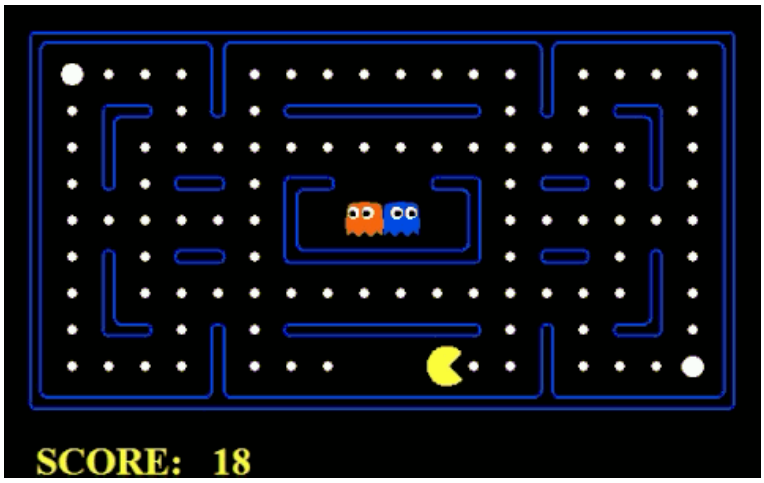
b) Task 2 (4.0 points): pacman

Students implement search strategies to help the pacman to collect all food points and visit four corners of the maze in an arbitrary order.

Visual studio code is recommended. Requirements include

- Formulate the given problem in form of a state space, determine details of related components.

- Implement UCS and A* algorithms to solve the problem.



Source: https://blog.sciencemuseum.org.uk/pac-man-turns-40/

- Implement a complete program to execute designated algorithms, in which
  - Input: path to a layout file; name of the expected algorithm
  - Output: list of actions (North, East, West, South, Stop); total cost
  - The maze structure is as below
    - **%** → obstacles/walls
    - **P** → initial location of the pacman
    - **.** → food points, there could be multiple points
    - spaces → blank cells.
    - There are no ghosts in the maze.

```
%%%%%%%%%%%%%%%%%%%%%%
% %%          % %      %
%    %%%%%% % %%%%%% %
%%%%%%      P  %      %
%    % %%%%%% %% %%%%%
% %%%% %          %   %
%         %%% %%%   % %
%%%%%%%%%    %%%%%% %
%.          %%        %
%%%%%%%%%%%%%%%%%%%%%%
```

  - Visualization: students visualize game steps on the console screen (press a key to start animation, do not need to press Enter for each step).

- For A* algorithm, students propose at least a heuristic function and determine its admissibility and consistency.
- Students organize the program regarding to the OOP model, ensure source code is compact and rational.

c) Advanced task (1.0 bonus point): Graphical Pacman

Given the source code folder **PacmanSearch**. Students conduct TODOs mentioned below to run the game with GUI.

TODO	*File*	*Content*
01	**search.py**	Implement a class to represent the problem in form of a state space. Related attributes and methods follow lecture theories.
02	**search.py**	Implement a class consisting of methods as search strategies. Organizing strategies in a single class is more professional than scattering them.
03	**searchAgents.py**	Implement a subclass of **Agent** (**game.py**). For each game step, **getAction**() function is called and the returned action is performed.

- Notes,
  - Related commands are listed in **commands.txt**
  - Layout files are provided in the folder **layouts**. Use a layout by passing its name, without the extension, to the command.
  - The default agent of the game is **KeyboardAgent** (the user control the pacman using arrow keys).

d) Task 4 (2.0 points): Presentation

- Student groups compose a presentation to report your work.
- **THERE IS NO PRESENTATION TEMPLATES. STUDENTS ARANGE CONTENTS IN A LOGICAL LAYOUT BY YOURSELVES.**
- The presentation must include below contents

- Student list: Student ID, Full name, Email, Assigned tasks, Complete percentage.
- Briefly present approaches to solve tasks, should make use of pseudo code/diagrams.
- Avoid embedding raw source code in the presentation.
- Study topics are introduced briefly with practical examples.
- Advantages versus disadvantages
- A table of complete percentages for each task.
- References are presented in IEEE format.

- **Format requirements:** slide ratio of 4x3, avoid using dark background/colorful shapes because of projector quality, students ensure contents are clear enough when printing the presentation in grayscale.
- Presentation duration is within **10 minutes**.

III. Submission Instructions

- Create a folder whose name is as

midterm_<group ID>

- Content:
  - **source** → source code folder. Students create a subfolder for each task.
  - **presentation.pdf** → presentation.
  - **demo.txt** → URL to the demo video with **the maximal duration of 03 minutes**.
- Compress the folder into a zip file and submit by the deadline.

IV. Policy

- **Student groups submitting late get 0.0 points for each member.**
- **Missing required materials in the submission loses at least 50% points of the presentation.**

- **Copying source code on the internet/other students, sharing your work with other groups, etc. cause 0.0 points for all related groups.**
- **If there exist any signs of illegal copying or sharing of the assignment, then extra interviews are conducted to verify student groups' work.**

-- THE END --