**Assignment 1**

# Relation to Course Learning Objectives

This assignment pertains to **CATEGORY 1 - General artificial intelligence objectives**:

**CATEGORY 1 - General artificial intelligence objectives**

1. Describe the principal features of intelligent behavior in terms of agent architecture and environment.
2. Explain, select and apply basic artificial intelligence techniques to solve simple or simplified problems.
3. Demonstrate an appreciation of the ethical issues that arise when decisions and actions are taken by inanimate agents, and the potential positive and negative impacts that such agents have on society.

**CATEGORY 2 - Topic-specific learning objectives**

1. ***Problem solving***: Use the general problem solving framework to describe a problem.
2. ***Search***:Select and apply different search strategies to find the solution to simple problems.

# Problems

1. **Agents.** For each of the five types of agents (simple reflex, model-based reflex, goal-based, utility-based, and learning), think of a real-world agent that fits that model (other than the ones given as examples in the book or slides). Provide a brief description of the agent and justify why it is a good representative of that type of agent.
2. **Environments**. We have characterized environments using seven different “dimensions” along which the environments can take different “values”. These include:
   1. Fully observable vs. Partially observable vs. Unobservable
   2. Deterministic vs. Strategic vs. Stochastic vs. Nondeterministic
   3. Episodic vs. Sequential
   4. Discrete vs. Continuous
   5. Static vs. Dynamic
   6. Single-agent vs. Multi-agent competitive vs. Multi-agent cooperative
   7. Known vs. Unknown

For each of the seven dimensions, pick a value. Then think of an environment that satisfies your choice, describe that environment and explain why the environment is representative of the dimension and the value you picked. E.g for dimension 5, select dynamic, then find an example of a dynamic environment and explain what makes that environment dynamic.

1. **Problem formulation.** Consider the “Pegs logic” game shown and described on the next page.
2. Describe a representation for a state of this game, and show how the **initial state** would look.
3. What are the **operator**s (**actions**) of this game?
4. Using C, C++, Java or Python, write the definitions for a state that you came up with in (a). Try to use a representation that would accommodate other game board shapes.
5. Using the state definition in (c), write the code for:
   * the **goal test** function
   * the **successor function** (transition model)
   * the **step cost** function (step cost is just 1 for this game, but it could be different)

|  |  |
| --- | --- |
| C:\Documents and Settings\Administrator\Desktop\ZZZZ\AUI\Courses\csc3309-5309_2012 Fall\Assignments\Assignment 2 - Problems & Games\VCS\pegs-logic-online-game-10.gif | The board at the left shows the initial configuration of the game, with 44 pegs (pink circles) arranged in the given configuration. The black circle in the middle is a hole with no peg in it.The goal of this game is to remove all pegs but one from the board. The perfect game leaves only one peg in the center position (the black one). Less perfect games leave one peg in other positions on the board.Pegs are removed by jumping over each peg with another peg. You can only jump over a peg if there is an empty space on the other side of it and you are right before it. E.g., in the picture to the left, only the pegs with circles around them can move. The peg they jump over is removed from the board. Only horizontal and vertical jumps are allowed, not diagonal ones.The game is over when no more jumps are possible. If more than one peg remains, the game is lost. |

1. Thinking **about search.** Consider the “Pegs logic” game in problem 3 above.
2. Analyze the branching factor of this game. Is it large? Can you reduce it? If so, how?
3. What search strategies that would be good to use to solve this puzzle? Why?

This assignment will develop into a programming assignment for CSC3309, but not for 5309.

# Grading

Scores will be assigned as shown in the table. This is individual work.

|  |  |  |
| --- | --- | --- |
| **PROBLEM** | **POINTS** |  |
| **Pb 1** | **10** | 2 points per type of agent |
| **Pb 2** | **14** | 2 points per type of environment |
| **Pb 3 Total** | **20** |  |
| **a)** | 3 | representation |
| **b)** | 1 | action |
| **c)** | 5 | code for representation |
| **d)** | 11 |  |
|  | *5* | code for goal test |
|  | *5* | code for successor function |
|  | *1* | code for step cost |
| **Pb 4 Total** | **6** |  |
| **a)** | 3 | branching factor |
| **b)** | 3 | search strategies |
| **TOTAL** | **50** |  |

# Submission

Turn in the the assignment to Jenzabar, under Coursework >> Assignment 1.

Your file should be called: “LastNameInitial\_Asg1.docx”. For example, “Cavalli-SforzaV\_Asg1.docx”.

I prefer .docx files to .pdf files. **Don’t submit a hard copy**, just the soft copy!

**Late penalty: 1% per hour late, rounded up to the next hour.**