Peer Evaluation Form – CSE 415 Project (Autumn 2019).

Team size (circle 1): 1 2 Team members (Last name, first name): _____

_____

Title: _____

Project Option for this project (circle one):

(1) Baroque Chess Agent.
(2) Wicked Problem Formulation.
(3) Backgammon Agent with Machine Learning.
(4) Hidden Markov Model algorithms and applications.

If Option==1: For each of the following: indicate whether you saw it or not by circling either "Saw" or "Didn't see".

1. (Saw, Didn't see). Play several moves of a game of Baroque Chess.
2. (Saw, Didn't see). Backed-up values from minimax search for the children of the current state.
3. (Saw, Didn't see). Evidence from the execution of number of ply used to determine most recent move. N_ply=___?
4. (Saw, Didn't see). Statistics about Zobrist hashing, such as how frequently a query for a state value succeeds.
5. (Saw, Didn't see). Evidence of Zobrist hashing being used to improve the number of cutoffs by alpha-beta pruning (e.g., a difference in number of cutoffs after turning on alpha-beta, but keeping the same max search depth).
6. (Saw, Didn't see). Evidence of superior play with good use of Zobrist, over no or just bad use of Zobrist hashing.
7. (Saw, Didn't see). Coherent personality of the agent, as revealed in the dialog.
8. (Saw, Didn't see). Some astute observation of the development or state of the game, as revealed in the dialog.

Were any other features shown? If so, what? _____

If Option==2: For each of the following: indicate whether you saw it or not by circling either "Saw" or "Didn't see".

1. (Saw, Didn't see). Clear explanation of what the problem is.
2. (Saw, Didn't see). Explanation of its wickedness in terms of some of the Rittel-Webber criteria.
3. (Saw, Didn't see). Rationale for modeling or simplification decisions.
4. (Saw, Didn't see). Session with the text-based interactive solving client.
5. (Saw, Didn't see). Run of a blind search algorithm (e.g., DFS or BFS) on the problem for some number of steps.
6. (Saw, Didn't see). Run of A* algorithm.
7. (Saw, Didn't see). Explanation of a heuristic for this problem.
8. (Saw, Didn't see). Presentation of some properties of the state space for this problem such as number of states as a function of search depth.
9. (Saw, Didn't see). Explanation of what would come next in order to make the problem formulation represent more of the wicked problem.
10. (Saw, Didn't see). Explanation of what was learned in the project.

Were any other features shown? If so, what? _____

If Option==3: Describe what form of machine learning was applied in this project: _____

1. Did you see V values or Q values for any particular states? _____
2. How many episodes of training did you see? (roughly) _____
3. Did you see the program EXPLOIT the results of learning? _____
4. Roughly how many distinct states are reachable from the initial state after the first player has completed his/her move? (ask the partners)_____
5. Did the agent achieve proficiency at playing Backgammon? (ask the partners if this is not clear) _____
6. How many training transitions and episodes were required to reach proficiency (i.e., optimal policy for the version chosen)? _____

Were any other features shown? If so, what?

If Option==4: What algorithms did you see implementations/demonstrations of?

1. Forward algorithm. 2. Viterbi algorithm (finding the most likely state sequence for a sequence of observations). 3. Other: _____

What dataset(s) were used during the demo?

Dataset 1: _____(Source: (a) online at _____ (b) made up. (c) other: _____
Dataset 2: _____(Source: (a) online at _____ (b) made up. (c) other: _____

1. Could you see the "emission" (observation) values as they were processed? Yes/No
2. Could you see the probability distribution for the current state ("belief"), at each point in the sequence? Yes/No
3. Could you see the most likely state sequence at the end of the Viterbi algorithm? Yes/No

For all 4 Project Options:

Implementation:

How complete is the implementation? (circle one) only some non-working code; some working, but incomplete code; basic working code; full prototype working.

Transparency: Circle those transparency features that are working: (1) textual display of intermediate results, (2) graphical display of current knowledge base, problem/game state, training data, or other data; (3) animation of updates to the graphical display; (4) other: _____

Interactivity: Does the project design include appropriate means for a user to interact with the program, so that the user has some control over what the program does, and so that the user can explore the way it works? _____

Presentation:

How many of the team members participated in the presentation? ____ / ___ (e.g., 2 out of 2)
Was an example or session discussed? _____
Was sample input shown? _____
Was sample desired output shown? _____
Was actual output shown? _____
Did you receive a clear explanation of how the featured AI technique works? _____
Based on that explanation, do you think you could effectively explain the technique to someone else?
Did you inspect some of the project's Python code? _____
Was it commented to explain how the main technique was implemented? _____
How clear were the comments? _____

Lesson(s): What did you personally learn from seeing this project? _____

Strengths (name at least one) : _____

Weaknesses (name at least one): _____

In what way(s) does or could the project better demonstrate the problem or technique it is addressing?
_____

Other comments: _____

Evaluated by (please print:) Last name:_____ First name:_____ (signed:) _____

Acknowledgement by project team: "I/we have read the above evaluation and I/we **agree/disagree** (circle one) with it."

Project member 1 signature: _____

Project member 2 signature: _____

(Optional rebuttal comments from the project team:)