Artificial Intelligence (CSE371) : Mini Project 1

Automated Game Playing (Adversarial Search)

> *"(games) offer a perfect laboratory for studying complex problem-solving methodologies. With a few parsimonious rules, one can create complex situations that require no less insight, creativity, and expertise than problems actually encountered in areas such as business, government, scientific, legal, and others. Moreover, unlike these applied areas, games offer an arena in which computerized decisions can be evaluated by absolute standards of performance and in which proven human experts are both available and willing to work towards the goal of seeing their expertise emulated by a machine. Last, but not least, games possess addictive entertaining qualities of a very general appeal. That helps maintain a steady influx of research talents into the field and renders games a convenient media for communicating powerful ideas about general methods of strategic planning."*
>
> *- Judea Pearl*
> *(Heuristics: Intelligent Search Strategies for Computer Problem Solving, 1984)*

This Mini Project deals with design of computer game playing tactics and strategy, and provides opportunity to learn search optimizations. You are required to program agents to play Backgammon. The first half of the project phase requires implementing a given tactic, while second half involves coming up with optimal and effective methods of your own. These will later be competed against each other in a tournament, with some additional scores to those who rise victorious.

Section 1: Backgammon (Rules and Gameplay)

(Note: find a familiarization desk HERE, ie. enjoy Backgammon here)

Find below, a detailed description of the traditional Backgammon Game which you will need to adapt in the following assignment:

Players
2 players.

Equipment

- Backgammon board
- 15 checkers for each player (traditionally, one set is tan and the other is brown)
- Two 6-sided dice

Goal

To be the first player to move all 15 of your checkers from their starting position to your home board .

Setup
Here is a feature that teaches you how to set up a Backgammon board.

Choosing the First Player (This is a function of the mediator)
Each player rolls a single die. The player with the higher roll goes first. (If there is a tie, the players roll again.)
The result of this roll is also used by the first player to make the first move of the game.

Moving Checkers

- On each turn, you first roll both dice. After rolling the dice, you move one or more checkers (if a legal move is available).
- The number rolled on each die determines how many points you can move. Your checker can only be moved to an open point. (An open point is one which is not occupied by two or more of your opponent's checkers).
- Each die constitutes a separate move. *EXAMPLE: If you roll a 4 and a 1, you can move one checker 4 spaces to an open point and a different checker 1 space to an open point, or you can move one checker 5 spaces to an open point. If you choose to use both dice for a single checker, an intermediate point (in this example, either 4 spaces or 1 space from the starting point) must be open.*
- You must always use as many of your die rolls as possible, even when doing so is not to your advantage. If only one legal move is available, you must take that move. If either move would be legal, but not both moves, you must use the higher number. If no legal move is available, you lose your turn.

NO Rolling Doubles to be included.

Hitting Your Opponent's Blot
If a single checker of either color is located on a point, that is known as a blot. If your checker lands on an opponent's blot, the opponent's checker is removed from the board and placed on the bar (the wooden area dividing the game board in half). This is known as "hitting" your opponent's blot.

Checkers on the Bar
If one or more of your checkers is on the bar, your must get those checkers back on the board before moving any others. This is done by moving the checker from the bar to an open point on your opponent's home board, corresponding to one of the numbers you roll. If both numbers rolled correspond to points which are not open, then you lose your turn.
If you can enter one or more of your checkers from the bar, but not all of them, you must do so. You then lose any remaining moves.

After your last checker has been returned to the board, any remaining numbers on the dice must be played. You may move any checker, including one that was just returned to the board.

Winning

The first player to move all 15 checkers from their starting position to their home board wins the game..

Section 2: A Reasonably Intelligent Player (RIP)

This player can be coded by using the **Holding Game Strategy**:

- Plan on keeping a point in your control that is located highly in your opponent's board, usually a point in his inner board or the bar point.
- The 20 point or bar points are the best holding game anchors, as they provide maximum chances to hit your opponent as he brings his checkers closer to home. Points further back get much weaker.
- Another key strategic element to the holding game is the distribution of the opponent's checkers. If he has only the 8 and 13 points made (as in the starting position), he will often have to leave a shot as he brings his checkers around. If he has made additional landing points in his outer board, your hitting chances go down significantly.

Section 3: Deliverables and Specifications

Submission 1: Submit an implementation of the RIP bot by 20 Jan - 11:59pm
Submission 2: Design a game-playing strategy which defeats your earlier RIP bot 24 Jan - 11:59pm
Tournament : TBA after Mid1

Constraints:
Maximum Number of checkers on each position can be 15

Your Home board:
If you are "Alice" , your home board should be A7-A12
If you are "Bob", your home board should be B7-B12

A1	A2	A3	A4	A5	A6	Z	A7	A8	A9	A10	A11	A12

B1	B2	B3	B4	B5	B6	Z	B7	B8	B9	B10	B11	B12

Input:
Alice
A1 A2 …. A12
B1 B2 …..B12
R1 R2
bar_variable (position of the bar point should be denoted by Z in the output)

Where A1-A12 and B1-B12 denote the state of the board (wrt each player) . A positive integer refers to that many white markers on the column, while negative denotes brown markers.
R1 and R2 are results of dice roll, ie. an integer between 1 and 6, uniformly at random.
The bar variable will be an integer which will denote the number of checkers for a player on the bar position. EXAMPLE : if player white (+ve) has 2 checkers on the bar, then the value of the bar_variable = +2.

You have to write your program from the perspective of Player 1 (ie. white markers).

Output:
S1 D1
S2 D2

Where, S1, D1, S2, and D2 denote columns, and S1 -> D1 and S2 -> D2 are the player's two moves, in that order.
Please note the moves are sequential, so if the output is of the form:

B1->B3
B3->B4

The mediator will first execute B1->B3 and then execute B3->B4.

Conditions for output:

- If any move is invalid please output the "pass" keyword for that move. EXAMPLE : If you want to move one checker from A1->A5 and the other to be passed your output will be:

  A1 A3
  pass

- If you want to move your checker from the bar to any position(say A2) and the other checker from B1->B3 your output will be:

  Z A2
  B1 B3