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.

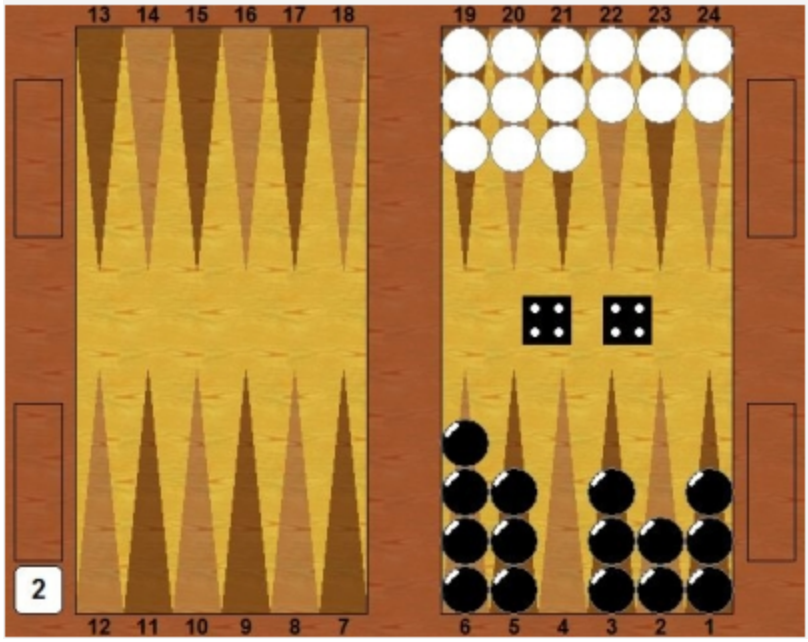
Hitting Your Opponent's Blot
If a single checker of either color is located on a point, that is known as a blot. If your checker lands on an opponent's blot, the opponent's checker is removed from the board and placed on the bar (the wooden area dividing the game board in half). This is known as "hitting"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

Bearing off

Your typical bearoff scenario is when there is no further contact (hitting) in a game, both players are racing to get their checkers off and of course you try to get off as many of your checkers as possible with the dice numbers rolled. So, bearing off would typically start at a scenario like this :



Winning

The first player to move all 15 checkers from their starting position off the board by bearing off wins..

For detailed rules refer to THIS.

Section 2: Player to ideally defeat a RIP

This player should ideally be strategically advanced to beat a player coded by the holding game strategy in the first phase of the assignment.

You are free to use any strategies, or combination of strategies you deem fit to achieve this.

Section 3: Deliverables and Specifications

Submission Deadline : 25.01.14, 11.59pm

Deliverables: tar/zip folder containing your C code, README describing the strategies used and other relevant information wrt your backgammon bot.

Constraints:

- Maximum Number of checkers of a particular player on any position can be 15 i.e a player can place all his checkers on a particular position.
- If the respective position on the board is empty it will be indicated by 0

Your Home board:

You are "Alice" , your home board is A7-A12
Opponent is "Bob", his home board is B7-B12

13	14	15	16	17	18	Z	19	20	21	22	23	24
12	11	10	9	8	7	Z	6	5	4	3	2	1

Input:
C1 C2 C3 …… C24
B
R1 R2

Where C[24] is denotes the state of the board, with positive entries denoting white markers, and negative denoting black markers. YOU ALWAYS HAVE TO PLAY ON BEHALF OF ALICE/WHITE, IE MOVING FROM LEFT TO RIGHT.
B is a string, consisting of 'a' and 'b', denoting the markers on the bar.
And R1 and R2 denote dice rolls, as performed by the mediator, with (R1, R2) being u.a.r. from Z6xZ6.

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:

1->3
3->4

The mediator will first execute C1->C3 and then execute C3->C4.

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 C1->C5 and the other to be passed your output will be:

  1 3
  pass

- If you want to move your checker from the bar to any position(say C2) and the other checker from C1->C3 your output will be:

  Z 2
  1 3

- For bearing off, from say C11, you output the move:

  11 O
  pass