CS154 PROJECT :

Topic :- Backgammon

Team Members:-

Rohit Kumar

Roll No. 120050028

Nitin Chandrol

Roll No. 120050035

A Brief Description :

Backgammon is a classical board game for two players, played according to the roll of dice and players win by removing all of their pieces from the board. It's strategy involves choosing the correct checker move available from numerous options **evolved by the dice roll, by anticipating possible counter moves by the second player.**

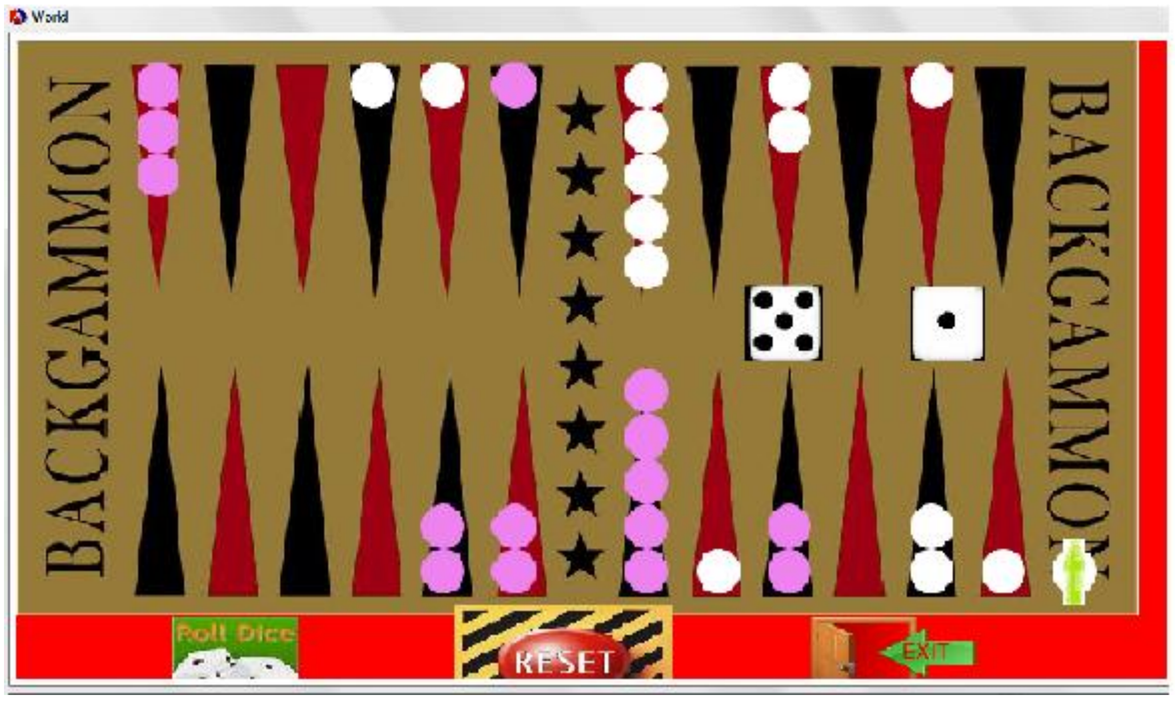
Overall Design Of The Game:

I. Determining whether a move is possible or not according to the set of numerous rules governing the game.

II. Among a set of possible moves determing the most efficient move taking account of the possibilities of   counter moves of opponent.

III. Evaluation of the static board value of a board taking account into various game factors like; checkers placed alone are non-preferable as they can be blotted (i.e sent back to starting position) by opponent's checker and vice-versa a move is preferable if it blots opponent's checker.

IV.   Replacing of checker and resetting of board and looping the process until someone wins.



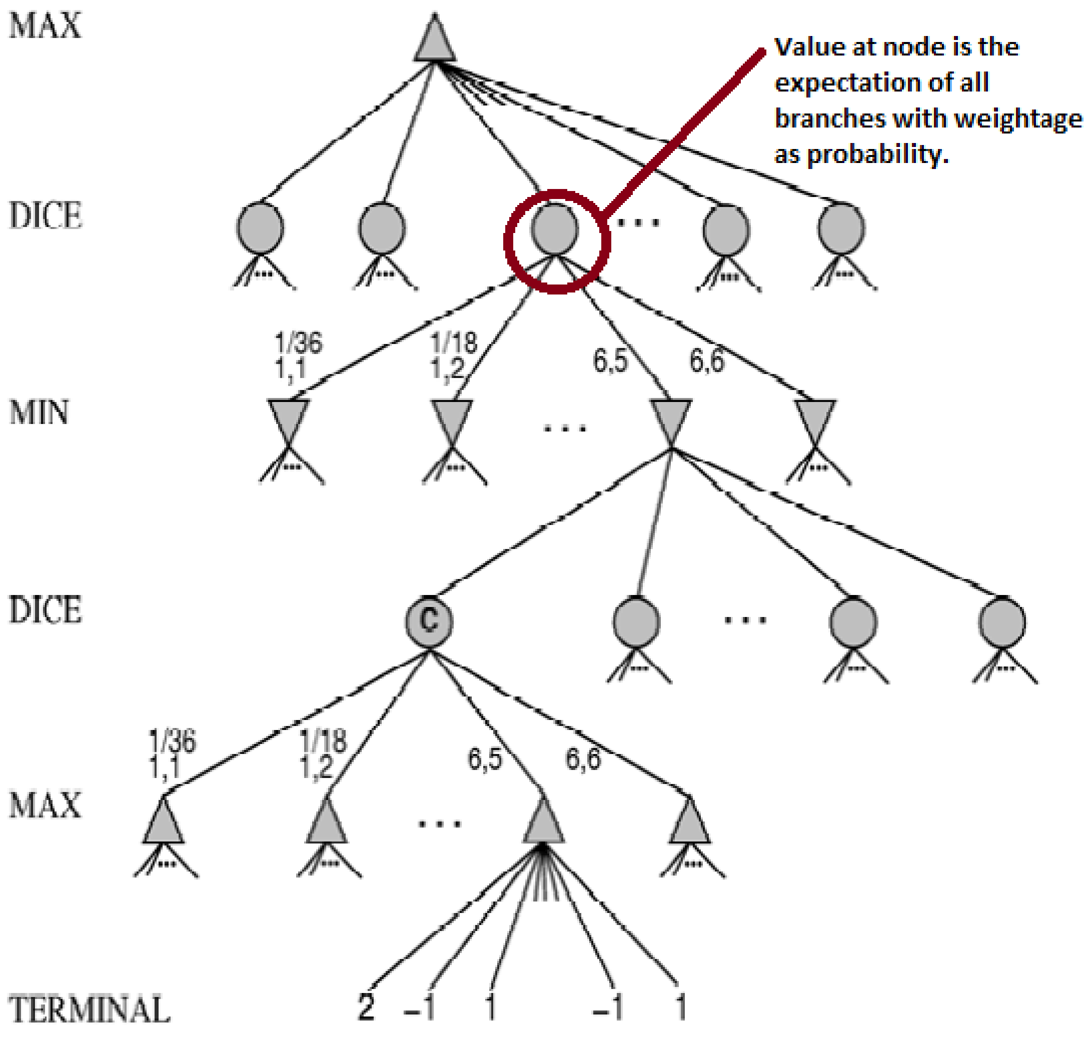
An Instance of the Game

Description of Algorithms and problems we faced:

I) EXPECTIMINIMAX ALGORITHM :

As, Backgammon is a combination of chance and strategy, so in addition to the minimax algorithm which maximizes player's gain and minimizes opponent's gain, a chance element is added at each node which evaluates the expectation value of the random event occuring corresponding to each random dice value.

The basic structure of algorithm is described as:


MAX
DICE
MIN
DICE
MAX
TERMINAL
Value at node is the expectation of all branches with weightage as probability.
1/36 1,1
1/18 1,2
6,5
6,6
C
2 −1 1 −1 1

II) STATIC BOARD EVALUATOR :

Our static board evaluator is much complex than what is generally described in zero-loss games due to the following aspects :-

a) We have given "weightages" to different situations involved, e.g. while calculating board value, we have evaluated, probabilistically the "chance" of getting blotted and the chance of blotting opponents checker.

b) On the basis of game-state we have assigned weightages in such a way that depending on different situation, AI adopts a defensive or an attacking strategy. This is done on the basis of board-state i.e. it depends on the indices where the player's and computer's checkers' are.

III) Alpha-Beta Pruning :

In effect with the expectiminimax algorithmn, increases the efficiency of evaluation by reducing the part to be evaluated.

IV) GRAPHICS And SOUND (too):

Used 'racket/gui’ to implement user interactive windows and the libraries "universe" and "image" to enable instruction inputs by mouse click events. It also uses imported image files(.png) and sound files in creating frames.

- One major problem faced by us was to find the next-correct state of the computer, as it needed to keep the correct state fixed and just virtually changing the states. As, simply bounding a temporary variable to a vector and changing it by using vector-set!, also changes the original vector due to having same pointers. We used vector-copy function to do the same.
- As, Backgammon is based on a heavy number of rules, it required a lot of networking among functions. We played the game numerous times to ensure that the game is prior to the set of rules.

Limitations and Bugs in program :

- For a given pair of dice, firstly the move is made in accordance to the maximum value of dice and then to the minimum of dice (in case both are equal, it does not matter). Whereas in reality the ordering of the moves doesn't matter and any move can be made.
- When dice values come to be equal then, 4 moves can be made (rather than 2). Due to complexity of the tree we were unable to make this in account.
- The blotted checkers are kept at the middle bar in actual board while we have kept it at rightmost position to avoid cumbersomeness of board.

Conclusions :

Backgammon has a lot of invariants available like "doubling cube" which speeds up game play which can be added.