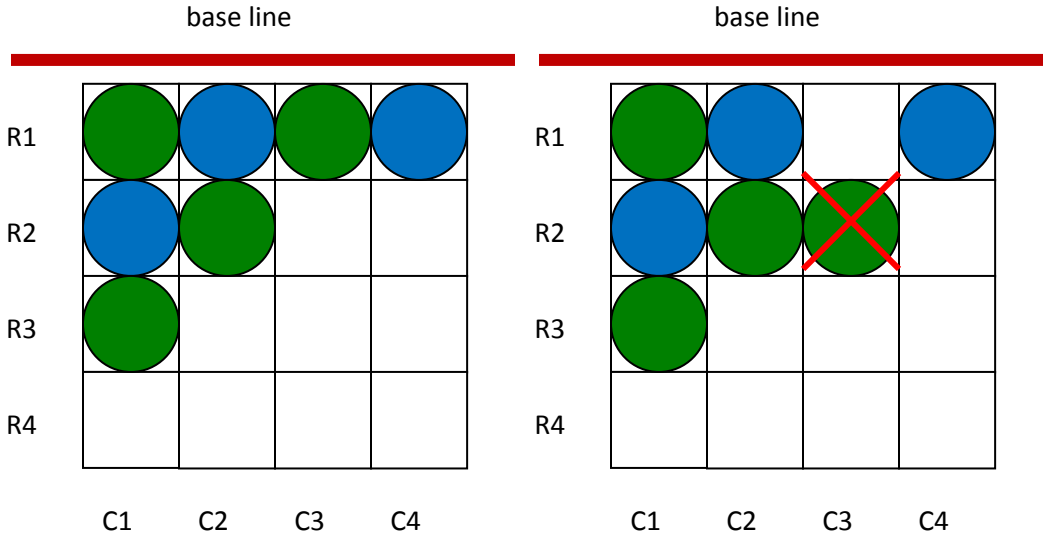
Programming Assignment 3: Game Playing

Problem Description

A 4x4 board game is played by two players (Fig. 1a). The game is deterministic, turn taking, zero sum game and can be named as ***align3***. The board positions can be specified by the row numbers R1, R2, R3, and R4; and column numbers C1, C2, C3, C4. The row R1 is nearest to the base line as mentioned in Fig.1a. The two players are you (human, say H) and the intelligent agent (machine, say M). In this game, player H gets 8 blue colored coins and player M gets 8 green colored coins. A player places a coin in any one column at a vacant row position while it is necessary to ensure that rows near the baseline should not be left vacant before a coin is placed (Fig. 1b). The two players take turn one by one and strategically place their coins to win. The player who achieves the alignment of the three coins wins while the alignment can be horizontal, vertical or diagonal (Fig.2). A single alignment of three coins in any direction by a player terminates the game. In this game, the coins never move once placed. If no player can achieve an alignment of three coins and all 8 coins of both players are consumed, then the game is a draw.



(a) *Feasible state*

(b) Infeasible state (player M places the coin wrongly in R2 in column C3, away from baseline)

Fig. 1: *align3* board

Utility values of the terminal states are as follows

+1 for WIN,

0 for draw

and -1 for LOSS

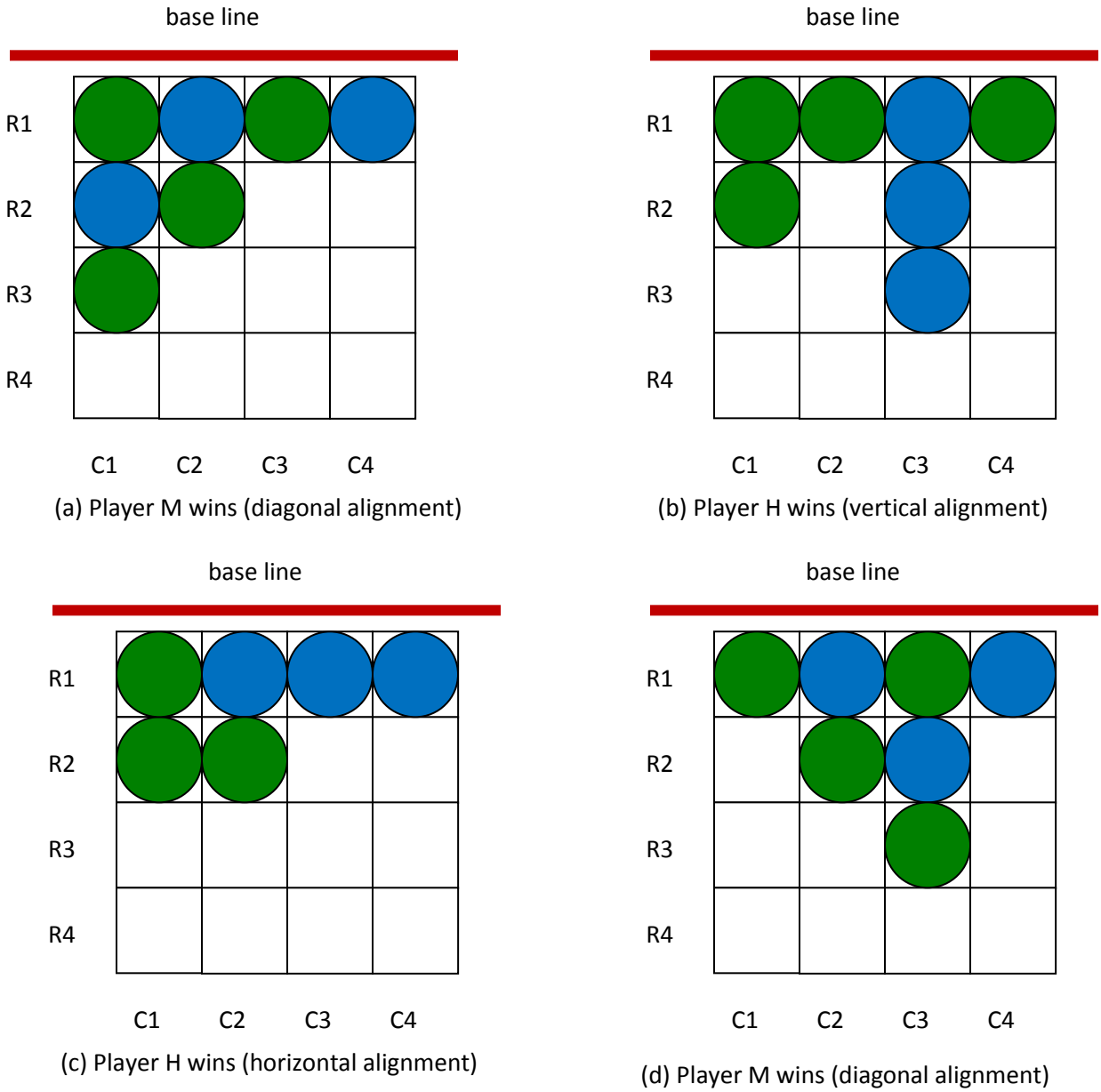


Fig. 2: WIN states

Represent the problem as state space search problem to compute a strategy for the intelligent agent (player M) to win. The initial state of the problem is an empty board and a move (action) is defined by placement of a coin on the board. A player has only four possible choices (one of the four columns C1-C4) for placing a coin on the board. The player does not have a choice of row as it is determined by the current state of the board already having other coins near the baseline (Fig 3). Therefore the branching factor is 4 for the above problem. A strategy consists of finding the initial move for M, next move of M based on a move by player H, next move of M based on another move by player H, and so on till the end of the game. As in Fig.3, you can visualize that if player H (or even player M, whose so ever turn it is) places the coin strategically in column

C2, wins. Identify the terminal states and associate utility values. Implement Minimax algorithm and Alpha Beta pruning to solve the above problem.

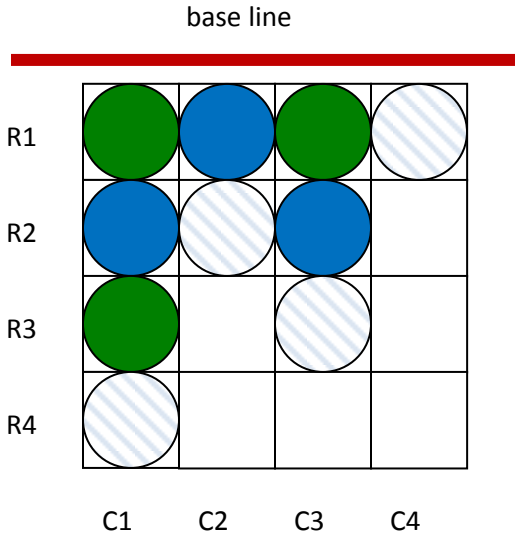


Fig. 3: Possible choices for a player (shaded)

Implementation

Use Python 3.7 (Windows 10), Turtle, PyQT and matplotlob for implementing your solution. Only standard Python libraries should be used. Support from external sources or libraries such as github will not be accepted in your submissions. Each student must design own solution and write own code. [Refer handout to understand the malpractice policies.]

Modules

Implement the following modules. The names are self explanatory and refer to the class discussions.

1. Successor_function(state s) returns a state.
2. Terminal_test(state s) returns a boolean value.
3. Utility_value(state s) returns a number.
4. MIN_VALUE(state s) returns utility_value.
5. MAX_VALUE(state s) returns utility_value.
6. Minimax algorithm
7. Alpha Beta pruning

Graphics

The game is expected to be played by the machine with a human player. Produce the graphics as shown in figures above, calibrate the upper left corner of the board and use cursor position to pick up the move by human player, use that as input to generate the strategy for player M's next move till the termination of the game. Turtle or PyQT graphics can be used to display the board and the coins. Also illustrate the WIN or LOSS of player M appropriately, or illustrate the draw

on the screen. Prompt the player H (human) for a new game once the previous game is over. If the player H wishes not to play, exit. List all the computed values of R1- R12 on the left hand side of the board.

Analysis Module

Produce the following analyses and display the resultant values.

(a) Minimax algorithm based analysis

i. Compute the number of nodes generated till the problem is solved. **[R1]**
ii. Compute the amount of memory allocated to one node. **[R2]**
iii. Compute the maximum growth of the implicit stack (if recursion is used) or of explicit stack used with the search tree. **[R3]**
iv. Compute the total time to play the game. **[R4]**
v. Compute the number of nodes created in one micro second **[R5]**

(b) Alpha Beta pruning based analysis

i. Compute the number of nodes generated till the problem is solved. **[R6]**
ii. Compute the ratio (R1 - R6)/R1 as saving using pruning. **[R7]**
iii. Compute the total time to play a game. **[R8]**

(c) Comparative analysis

i. Compare the memory used in both the techniques (Minimax and Alpha Beta pruning). **[R9]**
ii. Play the game 10 times and compute average time to play the game **[R10]**
iii. Play the game 10 times and compute the number of times player M wins **[R11]**
iv. Repeat (iii) 20 times and compute the average number of times player M wins. **[R12]**
v. Compare R4 and R8.

Driver

The driver must integrate all functionalities and execute the functions appropriately using interactive graphics display as described above. Use selection buttons for heuristics and also receive input from the human player appropriately . Show all pre computed values R1 to R12 and show the live game between your intelligent agent and human player. Also provide ways to receive input for executing your code at my end as well.

.

Writeup, evaluation and submission

Write up details will be made available two days before the submission. Evaluation will be out of 16 marks (8% weight). Students are advised to inform me immediately if any discrepancy exists

in this document. The assignment is due for submission on October 17, 2019 (Thursday) by 7:00 p.m.

All students are advised to work out the details of the solution and plan its implementation appropriately. Timely start, sound theoretical understanding of the subject, regularity in attending lectures, sincerity in applying the learned knowledge in problem solving, doing home work in time and an interest in the subject are the key parameters which contribute in the comfortable and smooth completion of the work given as assignment.

Students are advised to work out the details of the solution, discuss with me their complete understanding of the concepts individually and plan the implementation appropriately. In case of any difficulty, please feel free to contact me.

Vandana
October 7, 2019