

PROJECT REPORT: CONNECT 4

PROBLEM DESCRIPTION:

In this project, we tried to make a program that plays the game Connect 4 against human. We have also added the functionality to play the game in 2-player mode.

The game is a relatively simple abstract strategy board game where the players put coins of a color assigned to them into a 6x7 vertical grid. On dropping a coin into a column, the coin falls into the lowest available slot.

Objective of the game is to make a line of 4 or more of the player's coin, horizontally, vertically or diagonally. First player to make such a line wins.

PROGRAM DESIGN:

In our program, the user is able to make his move by clicking anywhere on the column he want to drop his coin in. Then the computer will try to calculate its best move and then make that move.

For enabling graphics in our program we used "Universe" library provided in DrRacket. The game board is designed based on the original Connect 4 game.

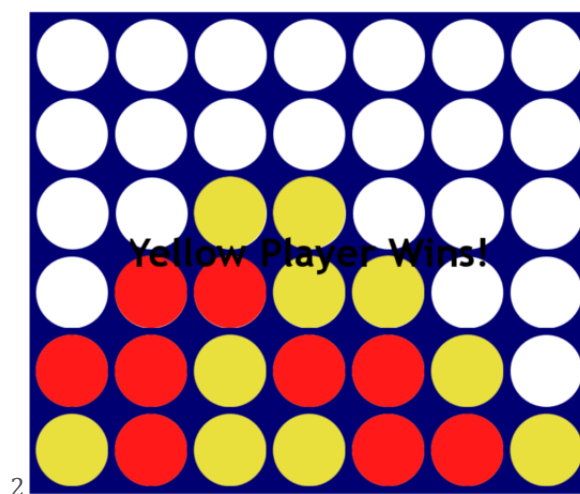
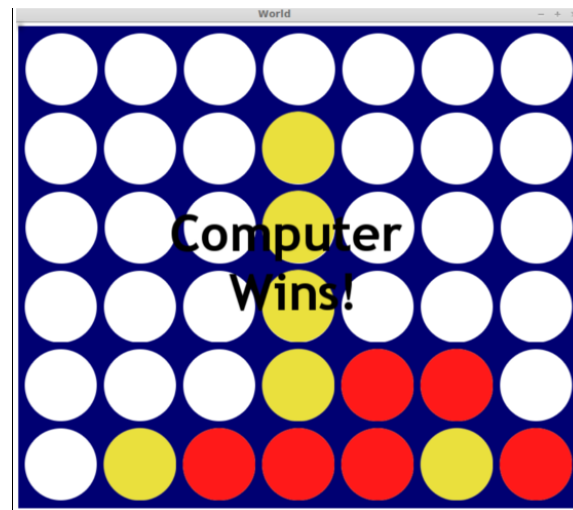
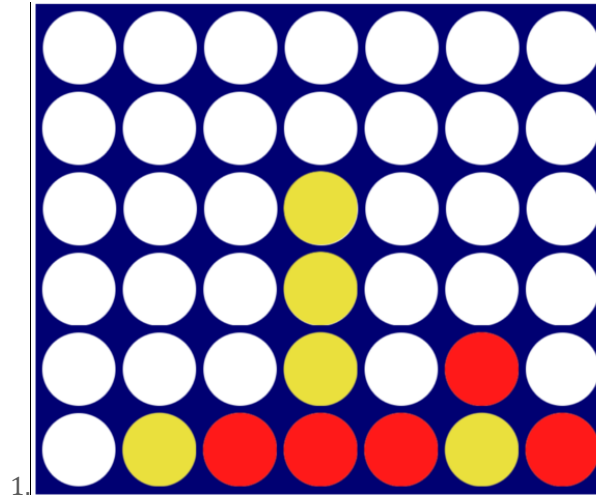
To calculate the optimal move, we used MiniMax Algorithm optimized by alpha-beta pruning. The algorithm was able to calculate all possible cases of the next four moves without delay in the computer we used.

Since it is not possible to calculate moves till the end of the game, we used a heuristics function to calculate the value of board position at the end of four moves. The function (score) calculates how good the game position is for computer by counting the number of threes in a row and twos in a row if the position is not a win or lose. The function then returns a value which is the sum of number of threes and twos after being multiplied by their respective weightages. In case three coins make a line but they can't be extended to four, the three won't be counted and same will be the case with twos. In case, the board position is a win for computer, the function will return a huge number (100000000). But doing just that won't distinguish a win three moves away and a win the immediate next move, so we multiplied the large number with what we named depth so that a win deeper in tree gets a lower priority. This ensures that program does not make a dumb move just because it expects a sure lose in future and that even if there's a sure win in future program will try to win at its first opportunity.

However, all the possible board states need not be calculated as some can ruled out (not to be the best) by using alpha-beta pruning. This

makes the evaluation of the best move for the computer faster. After each player plays few moves the game either ends as a win for either or a draw.

SAMPLE OF INPUT AND OUTPUTS:



LIMITATIONS AND BUGS:

This program limits itself to guess only to coming four or five moves hence it does not ensure a sure loose for the player.

The heuristic function that we have written does not specifically consider complex strategies/tactics.

OTHER POINT OF INTERESTS:

The computer can be made to play the game better if we increase the depth of minimax tree, but this will result in a delay.

Connect 4 is a solved game, i.e. if given enough memory and computation time and computer gets to play first, it can force a win. But it is not possible with the machines we have.