Intro to AI

Lab2

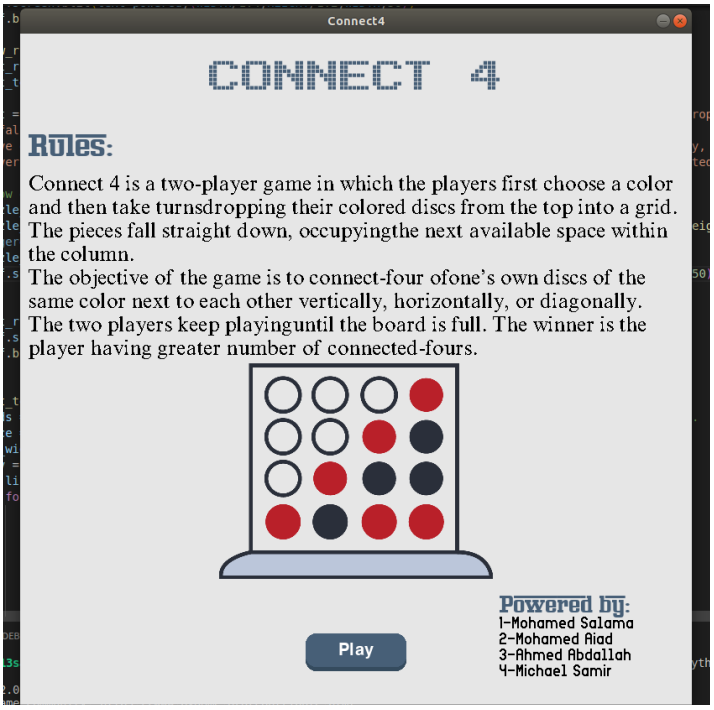
Name	ID
Michael Samir Azmy	19016237
Ahmed Abdallah Abo El Eid	19015274
Mohamed Momen Salama	19016474
Mohamed Tarek Hussein Aiad	19016405

Problem Statement: -

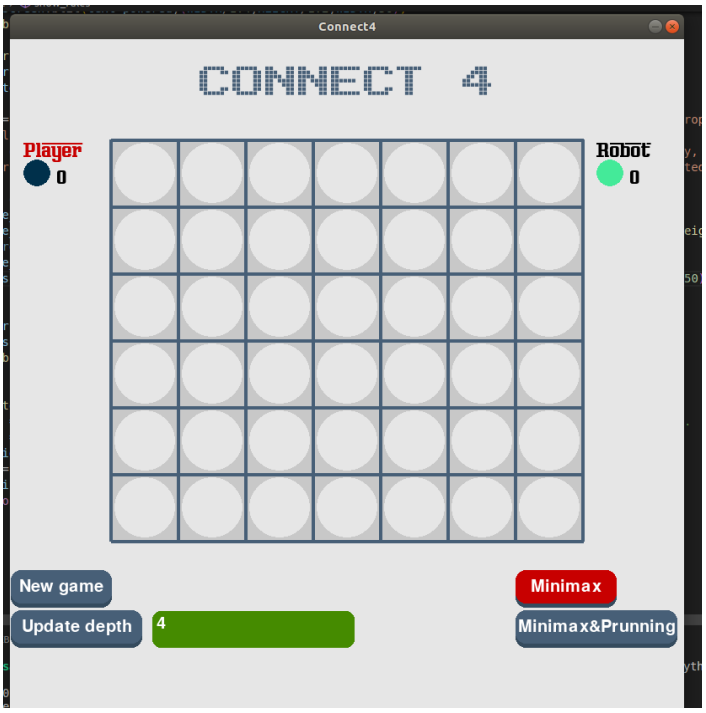
Playing connect 4 game until the board is full. The winner is the player with the greater number of connected-fours. using two algorithms (minimax - minimax-pruning).

User guide: -

1) When you start the program this page will appear.



2) After clicking play button Game will appear



3) you can determine depth from green label and click button “update depth” , to put circle in any column you want , you can click where you want to do it by mouse.player which has turn his word above will color “red”
4) score update continously while two players playing
5) score continue until the board is full.

Pseudocode:-

Minimax:-

```
minimax(state,max_height,cur_depth,isMax){
    if(cur_depth = max_height) return heuristic(state)

    if (isMax) {            // the robot turn
        value = negative infinity
        for (i = 0 to 7){       // check evey state
            if (the cloumn is empty) continue

            next_state = insert to this column
            temp = minimax(next_state,max_height
                                ,cur_depth+1,False)
            value = max(value,temp)
        } // end for

        return value
    } // end if
    else {                  // the player turn
        value = infinity
        for (i = 0 to 7){       // check evey state
            if (the cloumn is empty) continue

            next_state = insert to this column
```

```
            temp = minimax(next_state,max_height
                                    ,cur_depth+1,True)
            value = min(value,temp)
        } // end for

        return value

    } // end else

} // end func
```

Minimax with pruning

```
minimax_pruning(state,max_height,cur_depth,isMax,alpha,
beta){
    if(cur_depth = max_height) return heuristic(state)

    if (isMax) {            // the robot turn
        value = negative infinity
        for (i = 0 to 7){       // check evey state
            if (the cloumn is empty) continue

            next_state = insert to this column
            temp = minimax(next_state,max_height
                                    ,cur_depth+1,False)
            value = max(value,temp)
            if value >= beta
                return value
            alpha = max(alpha,value)
        } // end for

        return value
```

```
    } // end if
    else {                  // the player turn
        value = infinity
        for (i = 0 to 7){       // check evey state
            if (the cloumn is empty) continue

            next_state = insert to this column
            temp = minimax(next_state,max_height
                                ,cur_depth+1,True)
            value = min(value,temp)
            if value >= alpha
                return value
            beta= max(beta,value)
        } // end for

        return value

    } // end else

} // end func
```

Data Structures:-

1) list to store nodes expanded in bit manipulation to save memory
2) list to store heuristic value for each node

Assumptions:-

1) Heurostic
   loop on the board and find the number of connected 4, number of connected 3 and number of connected 2 for the two players and then substitute in this equation :-

10 * number of connect 4 of the player + 3* number of connected 3 of the player + number of connected 2 of the player - 10 * number of connect 4 of the computer - 3* number of connected 3 of the computer - number of connected 2 of the computer.

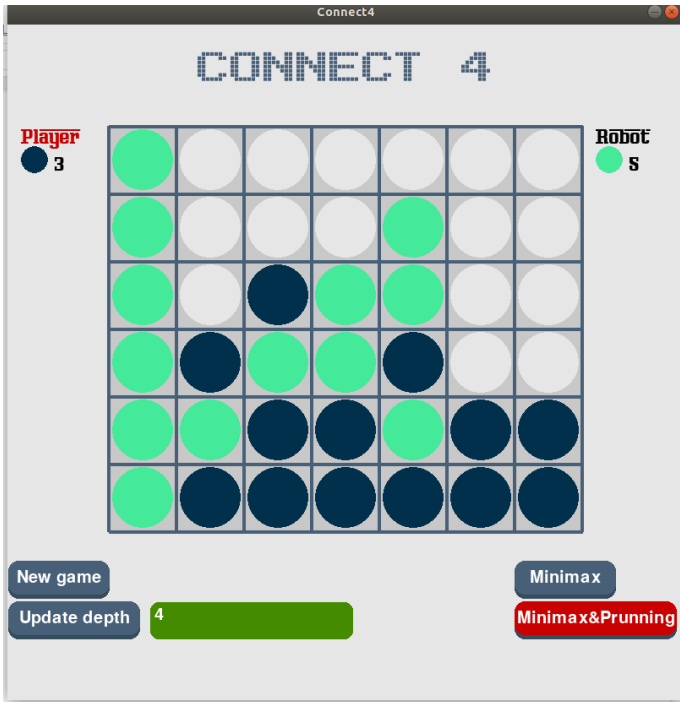
2) The board in the memory

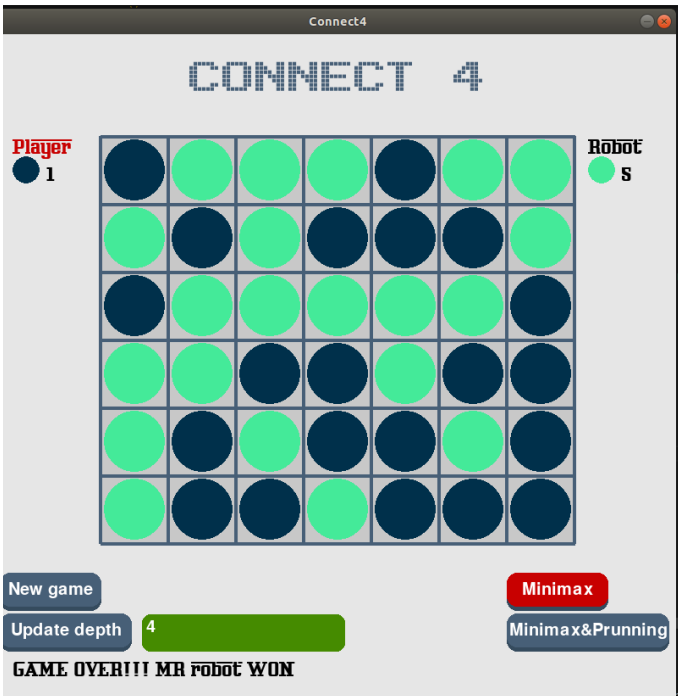
We saved every column in 9 bit so the final number of bits is 9*7 which is 63 bits (8 bytes).
The first 3 bits in every column is the first zero occurrence and the rest bits indicate that if 0 and it is less than the first occurrence of zero it is converted to 2.

Comparisons:-

	depth	Nodes-expanded	Time taken(s)
minimax	4	3200	0.256012201
minimax-pruning	4	324	0.0231297
minimax	5	22408	2.8673849
minimax-pruning	5	761	0.05423855
minimax	6	156822	18.647728
minimax-pruning	6	6994	0.3633933
minimax	7	1088297	130.9987
minimax-pruning	7	5975	0.33242464

Sample runs:-


Connect4
CONNECT 4
Player
3
Robot
5
New game
Update depth
4
Minimax
Minimax&Prunning


Connect4
CONNECT 4
Player
1
Robot
5
New game
Update depth
4
Minimax
Minimax&Prunning
GAME OVER!!! MR robot WON