

Bino Jacob

MCS 6833 – Concentration Project 2

Final Report



Contents

1.	Description	3
2.	Requirements.....	4
3.	Context Diagram	5
4.	Function Class List	6
5.	File Interaction Diagram.....	7
6.	Design.....	8
7.	Implementation Notes	9
8.	Test Cases.....	12
8.1.	Image Recognition Test Cases.....	12
8.2.	Game Play Test Cases.....	12
9.	Test Result and Summary	13
10.	References	16

1. Description

At public events there is a need to have activities where the audience can participate and wonder in the excitement of interact with scientific applications that are fun. This is the reason why I've chosen the vision based connect 4 game. This would be an ideal Computer Science project that does not seem to be performed before combining vision recognition along with robotic arm.

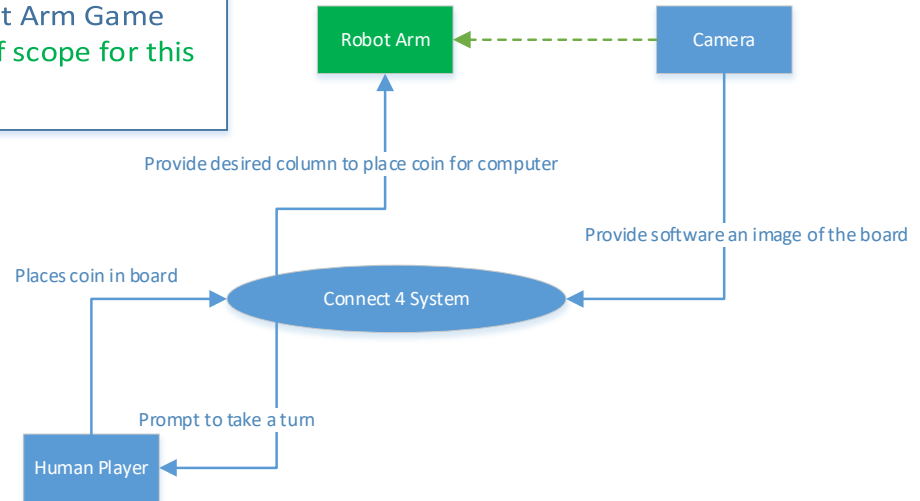
The purpose of this project is providing human players with an interactive game of Connect 4 where they compete with a computer player. The human player will simply need to interact with a game board while possibly being provided prompts from a computer screen notifying them to make a move or if the game is over. This has potential to be used in a public setting such as entertaining people in at events.

2. Requirements

1. The software shall identify where all the coins are placed in the board.
2. The software shall be able to determine if the human user has placed a coin in the board.
3. The software shall use an algorithm to determine the next position for the computer to place a coin in the board.
4. The software shall be able to determine if the computer user has successfully placed a coin in the board.
5. The software shall be able to differentiate between colors of the coins placed by each user and the game board.
6. After a player has made a move, the software shall determine if four coins are in a row in either a horizontal, vertical or diagonal direction anywhere on the 7 by 6 game board.
7. The software interface shall either prompt a human user to place a coin or notify the game is over.

3. Context Diagram

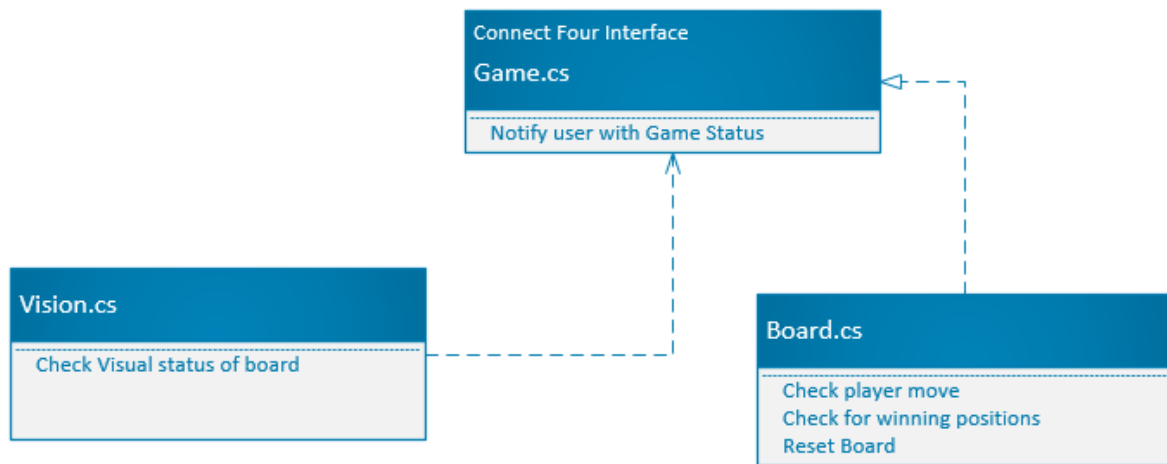
Blue Items are related to the Vision based Connect 4 Robot Arm Game
Green Items are out of scope for this project.



4. Function Class List

1. Drop coin for each player on Board
2. Check game status
 - a. Horizontal Case
 - b. Vertical Case
 - c. Diagonal (Top Left to Bottom Right)
 - d. Diagonal (Bottom Left to Top Right)
3. Reset Board
4. Check physical board with vision software

5. File Interaction Diagram



6. Design

6.1. Game board

It needs to process all game rules with the given values of a 2D array. After each player has inserted a coin, the game board will determine if there is a winner or the next move for the computer.

6.2. Computer vision

It needs to evaluate a physical connect game board and determine where the red coins, yellow coins and empty slots are. And it determines who the next player.

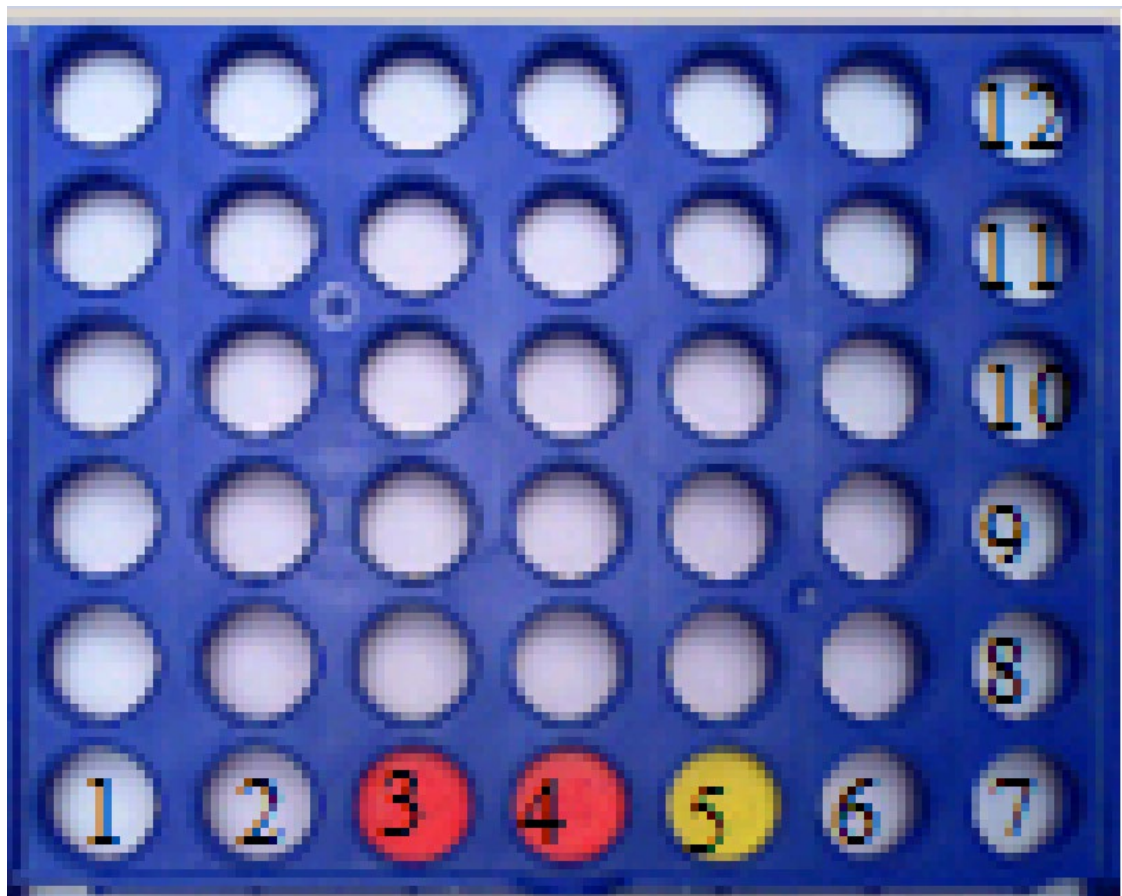
6.3. Graphical User Interface (for operator)

This will be the connection between the image recognition software and the game play software.

7. Implementation Notes

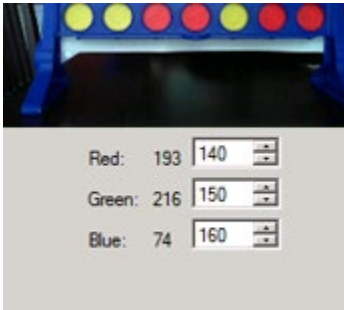
Analyze board positions

- Calibrate board with XY coordinates
 1. Select bottom row and first column as the first point and then continue selecting the remaining columns. Once the columns have been selected the X axis values are saved. Continue selecting the above rows to gather the Y axis values. Once all the XY points are gathered and the image has been clicked exactly 12 times, the RGB values will be read for each point.



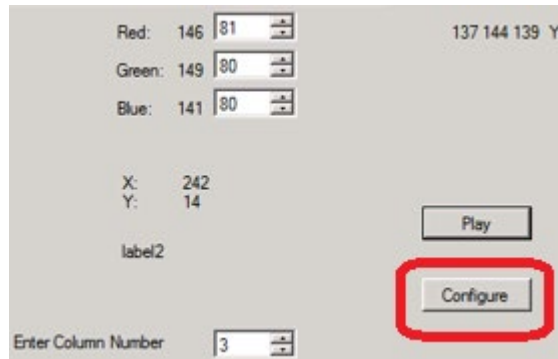
col1	col2	col3	col4	col5	col6	col7
123 120 130	149 146 156	151 150 151	Y	140 131 139	137 131 132	127 126 132
180 179 178	168 156 159	Y	Y	158 141 146	129 111 120	129 120 117
123 114 117	129 109 115	R	Y	142 112 119	131 110 114	129 113 113
112 101 104	R	R	R	136 107 109	132 109 111	124 111 110
R	Y	R	R	134 106 110	136 109 116	129 111 107
Y	R	Y	R	173 139 120	163 133 119	Y

- Initialize RGB values



Process human player moves and determine a move for the computer player

- Configure button allow operator of the software to determine if there are errors for the vision recognition filters.



- When game is ready to be played human player selects column on GUI, inserts coins into the game board and presses Play button.
- Utilize Minmax algorithm to determine column choice for computer player

In the case of the game, the maximizingPlayer would be set to false initially for player 2 (computer).

Pseudocode [\[edit \]](#)

The [pseudocode](#) for the depth limited minimax algorithm is given below.

```
function minimax(node, depth, maximizingPlayer) is
  if depth = 0 or node is a terminal node then
    return the heuristic value of node
  if maximizingPlayer then
    value := -∞
    for each child of node do
      value := max(value, minimax(child, depth - 1, FALSE))
    return value
  else (* minimizing player *)
    value := +∞
    for each child of node do
      value := min(value, minimax(child, depth - 1, TRUE))
    return value
```

```
(* Initial call *)
minimax(origin, depth, TRUE)
```

8. Test Cases

8.1. Image Recognition Test Cases

1. Test to differentiate between red, yellow coins and the board.
2. Test to ensure all pieces are correctly detected on the board.
3. Test different lighting conditions.
4. Test if human play only placed one coin each turn.
5. Test if robot arm has correctly placed coin on board in predetermined location.

8.2. Game Play Test Cases

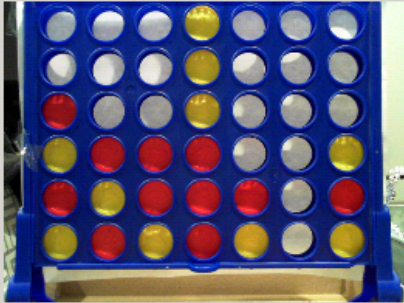
1. Test to determine if human wins.
2. Test to determine if computer wins.
3. Test to determine if there is a tie.

9. Test Result and Summary

The following results displays the results for testing game play and identifying game pieces.

Diagonal (negative direction) Win condition

Connect Four Vision Board



	col1	col2	col3	col4	col5	col6	col7													
143	162	162	167	172	175	165	167	170	Y	172	172	172	167	178	171	179	192	183		
215	220	206	219	221	210	225	225	217	Y	170	171	164	159	160	149	155	160	145		
R		146	145	145	155	154	148	Y		157	153	144	157	160	142	157	161	138		
Y		R		R		R				154	152	140	149	149	137	Y				
R		Y		R		R											156	155	135	R
Y		R		Y		R				Y							187	186	148	Y

Red: 162 81
Green: 163 80
Blue: 154 80

X: 216
Y: 16

label2

Play

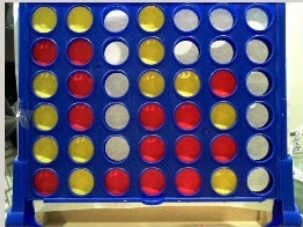
Configure

Enter Column Number 1

```
0002000
0002000
1002000
2111002
1211101
2121202
Player: 1 won Diagonal (negative)
```

Diagonal (positive direction) Win condition

Connect Four Vision Board



	col1	col2	col3	col4	col5	col6	col7									
Y	Y		190	194	192	Y	191	193	184	183	192	183	185	199	191	
R		R		234	238	214	Y	207	211	180	171	173	154	173	178	152
Y		R		168	169	150	Y	Y		R				169	175	142
Y		R		173	169	137	R		R	Y				164	170	135
Y		Y		163	163	130	R		R		R			167	173	135
R		Y		R		R			R	Y				181	187	142

Red: 145 81
Green: 152 80
Blue: 133 80

X: 216
Y: 15

label2

Play

Configure

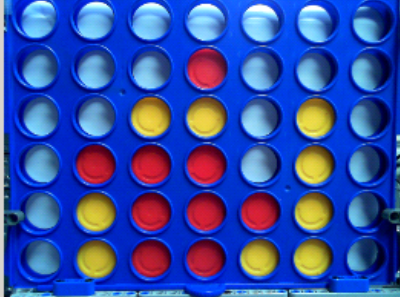
Enter Column Number 3

```
2202000
1102000
2102210
2101120
2201110
1211120
Player: 1 won Diagonal (positive)
```

Reset

Computer wins Vertical Win condition

Connect Four Vision Board



Red: 93 74
Green: 167 80
Blue: 202 80

X: 246
Y: 18

label2

Enter Column Number 2

Play

Configure

col1	col2	col3	col4	col5	col6	col7
95 170 223	110 188 233	113 184 232	118 188 239	120 199 250	71 139 228	0 49 183
88 153 200	114 183 223	123 193 226	R	113 180 231	91 170 231	16 49 173
103 169 208	116 176 213	Y	Y	137 205 237	120 194 228	22 71 182
101 164 202	R	R	R	125 189 222	Y	78 126 187
86 153 187	Y	R	R	R	Y	79 144 197
68 125 162	Y	R	R	Y	Y	54 118 188

```

0 0 0 0 0 0 0
0 0 0 1 0 0 0
0 0 2 2 0 0 0
0 1 1 1 0 2 0
0 2 1 1 1 2 0
0 2 1 1 2 2 0
  
```

Computer choose:6
Player: 2 won Vertical

Computer wins Horizontal Win condition

col1	col2	col3	col4	col5	col6	col7
112 161 226	100 138 196	132 177 218	147 187 227	149 193 226	134 184 215	118 166 196
101 129 159	118 150 187	131 167 197	128 156 190	130 163 196	125 162 188	123 171 191
Y	129 166 195	149 181 205	141 173 198	154 183 200	126 167 185	116 158 177
R	140 183 211	150 184 207	142 173 199	R	128 167 187	105 144 166
R	135 178 208	148 189 206	139 173 202	R	125 162 178	84 116 138
R	Y	Y	127 173 198	Y	109 148 166	44 66 97



Play

Configure

Reset

```

0 0 0 0 0 0 0
0 0 0 0 0 0 0
2 0 0 0 0 0 0
1 0 0 0 1 0 0
1 0 0 0 1 0 0
1 2 2 0 2 0 0
  
```

Computer choose:4
Player: 2 won Horizontal

Vertical Win condition

The interface displays a 7x6 Connect Four board. The left image shows the board with Red and Yellow pieces. The right image shows the board with Red and Yellow pieces, with a vertical win condition highlighted. The board state is as follows:

col1	col2	col3	col4	col5	col6	col7
174	180	187	169	169	169	181
181	181	181	187	187	187	188
188	188	188	183	183	181	159
168	163	144	144	142	137	138
131	163	161	151	151	148	137
156	155	151	159	160	152	147
149	138	158	160	149	141	141
127	R	Y	163	157	146	144
146	136	R	164	165	155	R
164	165	155	R	R	Y	165
162	140	153	155	140	Y	155
155	157	148	Y	R	R	155
147	132	141	145	123	Y	137
144	139	Y	R	R	R	Y

Red: 146 81
Green: 149 80
Blue: 141 80

X: 242
Y: 14

label2

Enter Column Number 3

Play

Configure

0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 1 2 0 0 1
0 1 1 2 0 0 2
0 2 1 1 0 0 2
0 2 1 1 1 2 2
Player: 1 won Vertical

Computer prevents human player from winning

The computer blocked the human player three times. But when the human player inserted the red coin in column 3, the computer determined that the human player could win either by placing a red coin in column 3 or column 6 so the computer determined it would lose and placed a coin anywhere.

The interface displays a 7x6 Connect Four board. The left image shows the board with Red and Yellow pieces. The right image shows the board with Red and Yellow pieces, with a diagonal win condition highlighted. The board state is as follows:

col1	col2	col3	col4	col5	col6	col7
Y	Y	190	194	192	Y	191
193	184	183	192	183	185	199
191	R	R	234	238	214	Y
207	211	180	171	173	154	173
178	152	Y	R	168	169	150
Y	Y	R	173	169	137	R
R	Y	Y	163	163	130	R
R	R	R	167	173	135	181
187	142	181	187	142	181	187

Red: 145 81
Green: 152 80
Blue: 133 80

X: 216
Y: 15

label2

Enter Column Number 3

Play

Configure

Reset

2 2 0 2 0 0 0
1 1 0 2 0 0 0
2 1 0 2 2 1 0
2 1 0 1 1 2 0
2 2 0 1 1 1 0
1 2 1 1 1 2 0
Player: 1 won Diagonal (positive)

10. References

https://en.wikipedia.org/wiki/Connect_Four

<http://blog.gamesolver.org/solving-connect-four/03-minimax/>

<https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-1-introduction/>

<https://stackoverflow.com/questions/36792847/implementing-and-using-minimax-with-four-in-row-connect4-game>

<https://stackoverflow.com/a/36802499>

- Utilized code from juharr (answered Apr 22 '16 at 20:08)

http://www.ifnamemain.com/posts/2014/Oct/09/csharp_connect4/

- Utilized code from this site as well

Utilized RGBclick code from Robotics Programming course.

<https://en.wikipedia.org/wiki/Minimax>