<table>
<thead>
<tr>
<th>CS 4100</th>
<th>comments</th>
<th>grades</th>
</tr>
</thead>
</table>
| 4773    | 1. Runs correctly. Can't beat it.  
2. Good optimizations | 100 |
| 8377    | No code submitted | NA |
| 9090    | 1. place your next assignment in a folder named 'assgn2'  
2. Computer overwrites my move and later on plays a turn without a move. | 90 |
| 1137    | 1. Works correctly in a few scenarios  
2. Good implementation  
3. Intuitive controls | 85 |
| 1623    | 1. Place your next assignment in a folder named 'assgn2'  
2. Readme and Doc missing. Didn't commit it?  
3. Can't see AI's move??  
4. Can beat the AI though it wins obvious games  
5. Not minmax, hardcoded moves.  
6. AI overwrites my move if i play 1,3. File showing this committed to repo.  
7. First move always hard-coded to center, because of this the AI isn't trying new moves. | 70 |
<table>
<thead>
<tr>
<th>1. Comp AI always starts (you commented out shuffle?)</th>
<th>4667</th>
<th>80</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Good heuristics. Comp AI is aggressive about wins.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| 1. First move always hard-coded to center, because of this the AI isn't trying new moves. | 3282 | 100 |
| 2. AI always wins |      |    |

| 1. Good implementation of min-max | 3951 | 100 |
| 2. You should probably make min and max the same function differing by an argument rather than 2 functions |      |    |

| 3. Nice GUI |      |    |

| 1. Comp overwrites my 2nd move when I start. Ex-

Me: 1,1, Comp:2,2, Me: 3,2... Computer overwrite 1,1

2. First move always hard-coded to center, because of this the AI isn't trying new moves.

3. Late by a day. (Last commit checked) | 0710 | 100 |

| 4. Good implementation of your own algorithm. Always wins. |      |    |

| 1. place your next assignment in a folder named 'assgn2' | 8755 | 85 |

| 2. The AI doesn't always make the best moves in order to win. But it always draws. |      | 100 |

| 1. skynet always atleast draws. |      |    |

| 2. Nice touch randomizing moves | 9838 | 100 |

<p>| 1. First move always hard-coded to top-left, because of this the AI isn't trying new moves | 1961 | 100 |</p>
<table>
<thead>
<tr>
<th>ID</th>
<th>Comments</th>
<th>Grade</th>
</tr>
</thead>
</table>
| 3080 | 1. Use SVN for the next assignment.  
2. The code does not run.  
3. Logically your code goes into an infinite loop |       |
| 3933 | No code submitted                                                                                                                                                                                         | 10    |
| 9977 | 1. Like you've said its a defensive AI that doesn't always try to win. With a few heuristics you could have changed that.  
2. 1 day late |       |
| 4133 | 1. 5 days late  
2. Could not run your code on my machine. Kindly demo it.                                                                                           | 95    |
| 3766 | 1. Be careful of indents. Stick to either tabs or spaces throughout. Use a good editor to review your code.  
2. Add '\n' to the end of a line if you continuing it on the next. Use "\n" in prints instead of literal new lines.  
3. Your program didn't run because you missed a ':' after else on line 192.  
4. Code allows me to play same move again. Ex moves (Comp plays first) - 1 8 8  
5. Comp overly defensive. Doesn't play obvious moves to win. Ex moves (Comp plays first) - 1 8 6. (it should have won by placing at 7 but plays 4. | 50    |
<table>
<thead>
<tr>
<th></th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Missing doc file explaining your algorithm</td>
</tr>
<tr>
<td>2</td>
<td>You should probably make min and max the same function differing by an argument rather than 2 functions</td>
</tr>
<tr>
<td>3</td>
<td>LookUp is a good idea for adding heuristics and for decreasing time taken on the first move. You could also check for rotations. This would save entries in the lookup.</td>
</tr>
<tr>
<td>8616</td>
<td>No code submitted</td>
</tr>
<tr>
<td>1</td>
<td>The AI is beatable when I start and consistently at that as there is no randomness so same fault can be exploited. Your pre_goal_test() is not doing its job. File showing this committed to repo.</td>
</tr>
<tr>
<td>0329</td>
<td>2. The AI draws/wins when it starts.</td>
</tr>
<tr>
<td>1</td>
<td>Place your next assignment in a folder named 'assgn2'.</td>
</tr>
<tr>
<td>2</td>
<td>1 day late</td>
</tr>
<tr>
<td>3</td>
<td>Make inputs from users case insensitive (for 'X' and 'O')</td>
</tr>
<tr>
<td>7281</td>
<td>4. Player selection should be random. This was a requirement. Comp AI never starts.</td>
</tr>
<tr>
<td>9856</td>
<td>No code submitted</td>
</tr>
<tr>
<td>1</td>
<td>AI does not finish the game when it starts, but gives me 2 (final) moves. File showing this committed to repo.</td>
</tr>
<tr>
<td>7730</td>
<td>2. AI can be beaten. File showing this committed to repo. The AI may be too aggressive when it's X?</td>
</tr>
<tr>
<td>1</td>
<td>AI can be beaten. Your algorithm accounts for my 1st corner or center move correctly, but not for my side moves. It doesn't consistently loose to the same ploy though as some corner moves can save it (you've randomized it). I believe your fourth move is the problem here. File showing this committed to repo.</td>
</tr>
<tr>
<td>6454</td>
<td>2. You should try using a depth search with your heuristic.</td>
</tr>
<tr>
<td>1257</td>
<td>No code submitted</td>
</tr>
<tr>
<td>9169</td>
<td>1. AI can be beaten. Your randomly choosing a move instead of doing a search to find a better move. File showing this committed to repo.</td>
</tr>
<tr>
<td>2651</td>
<td>No code submitted</td>
</tr>
</tbody>
</table>
1. Place your next assignment in a folder named 'assgn2' and not 'AI'... Let me know if you need help cleaning your repo.

2. AI can't be beaten. Good implementation of alpha-beta pruning.

<table>
<thead>
<tr>
<th>1264</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>No code submitted</td>
<td>NA</td>
</tr>
</tbody>
</table>

1. If you've implemented alpha-beta pruning then you shouldn't need to random-hard-code the first move. Because of that, your algorithm cannot be extended to other problems easily.

2. AI can't be beaten but it can make really silly moves (even thought it eventually wins). You should check shortest path to a win to fix this.

3. random.randint returns a random integer.

<table>
<thead>
<tr>
<th>7054</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>4. Good UI implementation</td>
<td></td>
</tr>
</tbody>
</table>

1. You haven't handled for an empty input from the user. This can be annoying if I strike "Enter" twice.

2. Hard-coding the first move decreases the generality of the algorithm.

<table>
<thead>
<tr>
<th>2177</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. You should probably make min and max the same function differing by an argument rather than 2 functions</td>
<td></td>
</tr>
</tbody>
</table>

1. Why does your display show (1-10) allowed as a move when there are only 9 blocks?

2. AI is beatable. You randomly select a move without looking at who could win 2-3 steps ahead.

<table>
<thead>
<tr>
<th>6568</th>
<th>90</th>
</tr>
</thead>
<tbody>
<tr>
<td>No code submitted</td>
<td>NA</td>
</tr>
</tbody>
</table>

1. Could not run your code on my machine. Please demo this code with me.

<table>
<thead>
<tr>
<th>1386</th>
<th>90</th>
</tr>
</thead>
<tbody>
<tr>
<td>No code submitted</td>
<td>NA</td>
</tr>
</tbody>
</table>

1. No code submitted | NA |

<table>
<thead>
<tr>
<th>1366</th>
<th>NA</th>
</tr>
</thead>
<tbody>
<tr>
<td>No code submitted</td>
<td>NA</td>
</tr>
<tr>
<td>ID</td>
<td>Comment</td>
</tr>
<tr>
<td>------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>6211</td>
<td>1. 2 days late</td>
</tr>
<tr>
<td></td>
<td>2. AI is unbeatable in min-max</td>
</tr>
<tr>
<td>1133</td>
<td>1. AI unbeatable.</td>
</tr>
<tr>
<td></td>
<td>2. Good simple implementation of alpha beta.</td>
</tr>
<tr>
<td>6015</td>
<td>1. AI unbeatable.</td>
</tr>
<tr>
<td></td>
<td>1. AI unbeatable.</td>
</tr>
<tr>
<td>6991</td>
<td>1. AI unbeatable.</td>
</tr>
<tr>
<td></td>
<td>2. Good design of classes</td>
</tr>
<tr>
<td>3567</td>
<td>1. AI is beatable. Your Min-Max algorithm has some flaw in it.</td>
</tr>
<tr>
<td>3144</td>
<td>No code submitted</td>
</tr>
<tr>
<td>9474</td>
<td>1. AI is beatable. Your Min-Max algorithm has some flaw in it.</td>
</tr>
</tbody>
</table>