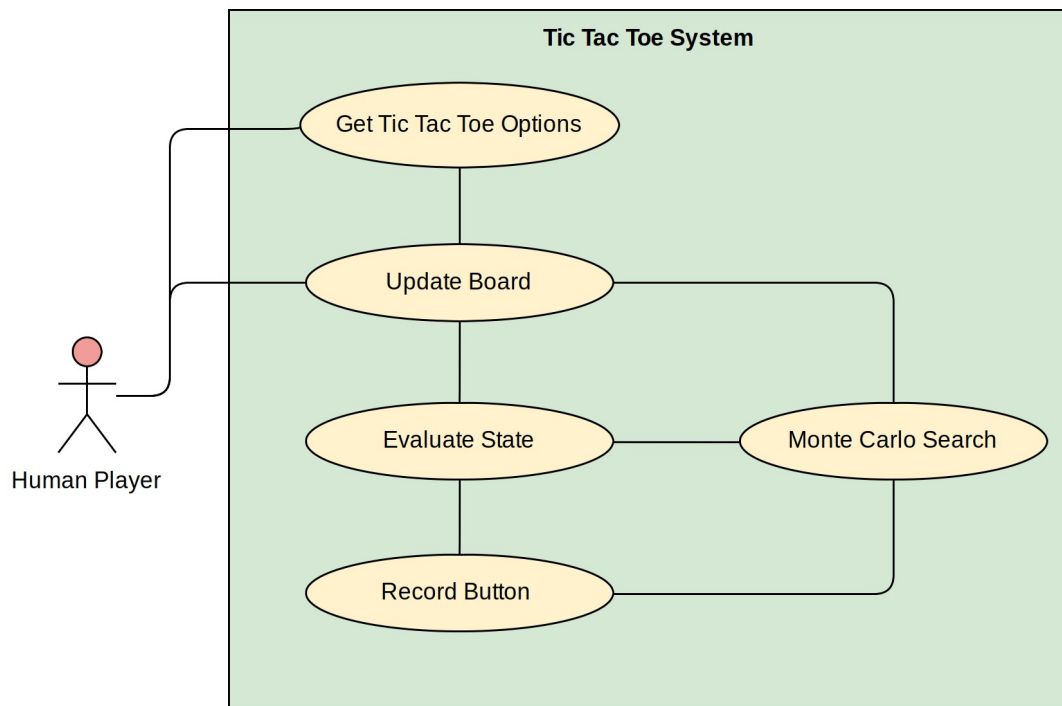


## Use Case Diagram



<i>Use Case Name</i>	Get Tic Tac Toe Options
<i>Participating actors</i>	Initiated by Human Player
<i>Flow of events</i>	<ol style="list-style-type: none"><li>1. The Get Tic Tac Toe Options is activated by the Human Player when the player's choices are entered.</li><li>2. The Human Player's choices are passed on to Update Board, which uses this information to find the next move.</li><li>3. Update Board gives information to Get Tic Tac Toe Options on whether the game has ended.</li></ol>
<i>Entry condition</i>	<ul style="list-style-type: none"><li>• The Human Player selects an option presented by Get Tic Tac Toe Options.</li></ul>
<i>Exit Condition</i>	<ul style="list-style-type: none"><li>• Human Player is informed if the game ended in a win, a loss or a draw.</li></ul>
<i>Quality Requirements</i>	<ul style="list-style-type: none"><li>• The move is instantaneously recorded and applied</li><li>• The game ending status is immediately applied when available, otherwise board information is updated after 1500 simulations.</li></ul>

<i>Use Case Name</i>	Update Board
<i>Participating actors</i>	Human Player makes moves on the board.
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. Update Board is activated when it is provided with information by Get Tic Tac Toe Options on the Human Player's choices.</li> <li>2. Update Board gets a Human Player move, and checks with Evaluate State if the move is terminal.</li> <li>3. If the move is not terminal, Monte Carlo Search is called.</li> <li>4. Monte Carlo Search informs Update Board on computer's next move.</li> <li>5. The computer's move is applied on the board by Update Board.</li> <li>6. Otherwise, Evaluate State informs Update Board that the game has ended.</li> <li>7. Get Tic Tac Toe Options is informed that the game has ended.</li> </ol>
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>• The Human Player has already activated Get Tic Tac Toe Options.</li> </ul>
<i>Exit Condition</i>	<ul style="list-style-type: none"> <li>• The Human Player is presented with a new move on the board, Or</li> <li>• The Human Player is informed that the game has ended.</li> </ul>
<i>Quality Requirements</i>	<ul style="list-style-type: none"> <li>• The move is instantaneously recorded and applied</li> <li>• The game ending status is immediately applied when available, otherwise board information is updated after 1500 simulations.</li> </ul>

<i>Use Case Name</i>	Monte Carlo Search
<i>Participating actors</i>	
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. Update Board calls Monte Carlo Search to evaluate the next move.</li> <li>2. Monte Carlo Search calls Record Button to determine the available squares on the board.</li> <li>3. 1500 simulations are performed on the available squares.</li> <li>4. During the simulation Record Button is called and UCT values of the squares are updated.</li> <li>5. Monte Carlo Search determines the square with the highest UCT value.</li> <li>6. Monte Carlo Search tells Update Board to select the square with the highest UCT value, and provides the square identity to Update Board.</li> </ol>
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>• Evaluate State cannot find an immediate terminal move.</li> </ul>
<i>Exit Condition</i>	<ul style="list-style-type: none"> <li>• Update Board gets the location of the next move.</li> </ul>
<i>Quality Requirements</i>	<ul style="list-style-type: none"> <li>• The move is instantaneously recorded and applied</li> <li>• The game ending status is immediately applied when available, otherwise board information is updated after 1500 simulations.</li> </ul>

<i>Use Case Name</i>	Record Button
<i>Participating actors</i>	
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. Evaluate State informs Record Button on the button which the user clicked, and its mark.</li> <li>2. Record Button stores this information and makes it available to Monte Carlo Search.</li> <li>3. Monte Carlo Search calls Record Button to determine the number of available squares.</li> <li>4. Monte Carlo Search calls Record Button to update its UCT scores.</li> <li>5. Monte Carlo Search calls Record Button to obtain the identity of the square with the highest UCT score.</li> </ol>
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>• Evaluate State calls Record Button to store information on a square.</li> </ul>
<i>Exit Condition</i>	<ul style="list-style-type: none"> <li>• Monte Carlo Search has a list of squares which are still available.</li> </ul>
<i>Quality Requirements</i>	<ul style="list-style-type: none"> <li>• The move is instantaneously recorded and applied</li> <li>• The game ending status is immediately applied when available, otherwise board information is updated after 1500 simulations.</li> </ul>

<i>Use Case Name</i>	Evaluate State
<i>Participating actors</i>	
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. Record Button passes a list of available unclicked buttons to Evaluate State.</li> <li>2. Evaluate State checks if immediate terminal moves exist.</li> <li>3. If terminal game ending moves exist, Evaluate State passes this information to Update Board.</li> <li>4. If no terminal moves exist, Evaluate State calls Monte Carlo Search to determine the next move.</li> </ol>
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>• Record Button has a list of clicked, and unclicked buttons.</li> </ul>
<i>Exit Condition</i>	<ul style="list-style-type: none"> <li>• Either Update Board selects a terminal square to end the game, Or</li> <li>• Monte Carlo Search starts performing 1500 simulations to determine the next move.</li> </ul>
<i>Quality Requirements</i>	<ul style="list-style-type: none"> <li>• The move is instantaneously recorded and applied</li> <li>• The game ending status is immediately applied when available, otherwise board information is updated after 1500 simulations.</li> </ul>