# Playing Tic Tac Toe with myCobot 280 M5

**Jay Menon**
School of Manufacturing Systems and Networks, Arizona State University, Arizona, USA

**Article Info**

***Keywords:***

Tic-Tac-Toe
Alpha Beta Pruning algorithm
myCobot

**ABSTRACT**

In this research, we present a novel application of robotics and simulation to the game Tic-Tac-Toe, wherein a robotic arm called MyCobot imitates the movements of a computer player. By employing the pymyCobot library and the Alpha-Beta Pruning algorithm, the research shows that virtual and physical entities can synchronize smoothly. The 3x3 grid-based MyCobot's movements replicate gameplay, demonstrating possible uses in interactive systems and human-robot interaction.

## 1. OBJECTIVE

This code aims to implement a basic game of Tic-Tac-Toe in which a human player (X) can compete against a computer player (O). Players move by choosing a spot on the board in turns as the game is played on a 3x3 grid. The Alpha-Beta Pruning algorithm is a minimax search algorithm tailored for computer games that determines the moves made by the computer player.

The code uses the pymyCobot library to control a robotic arm (MyCobot) in addition to the traditional Tic-Tac-Toe gameplay. The integration of robotic control and game logic is demonstrated by the robotic arm, which replicates the movements of the computer player on a physical board. To improve the game's interactive and demonstrative elements, the real robotic arm is moved to select and place locations on the board.

## 2. METHOD

To control the MyCobot robotic arm, the code first imports the required libraries, which include pymyCobot, time for managing time-related operations, os and sys for system-related functionality, and random for creating random moves in the game.

A class named Tic is defined to encapsulate the Tic-Tac-Toe game logic and control of the MyCobot robotic arm. This class includes methods for initializing the game board, MyCobot, displaying the board, checking game completion, determining the winner, and handling player moves. The Tic class defines the global variables and winning combinations for the game of Tic-Tac-Toe. The MyCobot instance is supposed to be stored in the mc variable.

The Tic-Tac-Toe board is initialized using the __init__ method. It initializes an empty board with None values if no custom board is supplied. The MyCobot instance is initialized by the jay_init method, which connects to the given port and sets initial angles. The Tic-Tac-Toe board's current state is printed using the show method. To check for move availability, determine game completion, and identify the winner, methods like available_moves, available_combos, complete, X_won, O_won, tied, and winner are defined.

To retrieve a player's squares, the method get_squares is defined. The MyCobot robotic arm moves in tandem with player movements thanks to the methods make_move and move_cobot. The MyCobot's suction mechanism is managed by the pump_on and pump_off methods. Based on the current state of the game, the

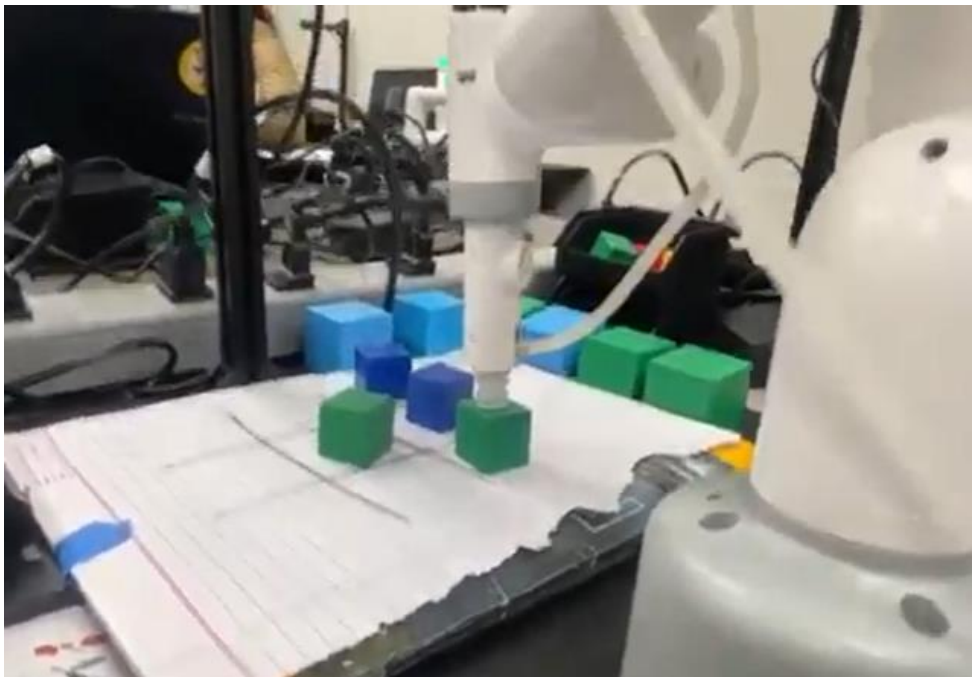
determine function uses the Alpha-Beta Pruning algorithm to determine the optimal move for the computer player.

The MyCobot robotic arm physically imitates the movements made by the computer player, showcasing an interactive integration of robotics control and game logic. The winner is determined at the end of the game.

## 3. CHALLENGES FACED

The suction pump was not working at times that leads to dropping off cubes even after the pump is logically on. The end effector loosens at times which results in wrong positioning of the cube.

## 4. MEDIA



The final video for the demonstration can be found here:
https://drive.google.com/file/d/14HsgitNgQSjTNtvcr3J9iy2a0O86VWoA/view?usp=sharing

The Final video presentation can be viewed here:
https://www.canva.com/design/DAF1-54YE1k/D3knHNssBOYpDwmuXXf8tw/view?utm_content=DAF1-54YE1k&utm_campaign=celebratory_first_publish&utm_medium=link&utm_source=editor_celebratory_first_publish

## 5. CONCLUSION

To sum up, this project effectively combines simulation and robotics by implementing a Tic-Tac-Toe game in which a robotic arm called Mycobot physically mimics computer player movements. The smooth synchronization between virtual and physical entities is demonstrated by the integration of the Alpha-Beta Pruning algorithm for strategic decision-making and the pymycobot library for precise robotic control. This work illustrates the potential applications in educational, recreational, and interactive settings in addition to offering an engaging and interactive demonstration of human-computer collaboration.

## REFERENCES

[1] Elephant Robotics myCobot M5 280, https://docs.elephantrobotics.com/docs/pdf/myCobot-en.pdf
[2] Sample Video Presentation, https://www.youtube.com/watch?v=HTXOMVjY4V4