

Cozmo Tic Tac Toe

Sean Kim (sdkim2)

Project Overview

- Main Goal:
 - Create a Tic Tac Toe game where the Cozmo robot plays against a human using GPT 4 for decision-making and image analysis.
- Detailed Objectives:
 - Cozmo will see a generated window screen with a Tic Tac Toe board.
 - Cozmo will play first as the piece 'O'.
 - Cozmo will announce its next move after the human makes their move.
 - If Cozmo detects that game has ended, it will prompt to reset game and announce winner.
 - The game follows the same rules as any other Tic Tac Toe game.

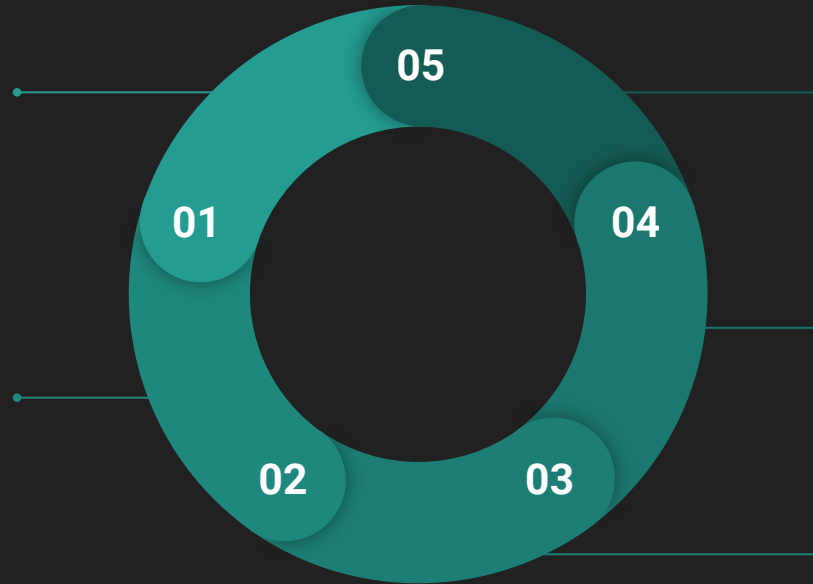
Code Flow and Approach

Image Capture (GetImage)

Capture an image of the Tic Tac Toe board using Cozmo's camera. The image is converted to JPEG format and encoded in base64 for analysis by GPT-4.

Image Analysis (RunQuery)

Send the captured image to GPT-4 with a prompt to analyze and determine the current state of the Tic Tac Toe board.



Get Move (ExtractIndex)

Extract the move index from GPT-4's response. Update the board state and instruct Cozmo to make the move.

Make Move (AskMove)

Ask GPT-4 for the best move based on the current board state. GPT-4 determines the optimal position for 'O' to be placed.

Board Parsing (ParseBoard)

Parse GPT-4's response to extract the board state as a list of characters representing 'X', 'O', and empty spaces.

Prompt Engineering: Getting the Board State

- Step-by-Step Approach
 - Step 1: Image Analysis
 - *“This image shows the Tic Tac Toe board. Analyze and return the board state.”*
 - Step 2: Board Parsing
 - *“Return just the board state by converting the description to a list of nine characters, where each character is 'X', 'O', or '-'. Make it one line please. For example : ['O', 'X', 'O', '-', 'X', 'X', '-', '-', '-]”*
- Benefits
 - Clear and focused responses
 - Minimizes the risk of errors in understanding and returning the correct board state

Prompt Engineering: Getting the Next Move

- Step-by-Step Approach
 - Step 3: Determine the Best Move
 - *“This is the current tic tac toe board: {format_tic_tac_toe(current_board)}, where is O's next best valid move?”*
 - Step 4: Extract Move Index
 - *“Return the integer index of where the description is saying it is best to place 'O', top left is 1, bottom right is 9.”*
- Benefits
 - Clear and manageable responses
 - Reduces errors by focusing on one task at a time

Limitations

- Analyzing Board State (Issue with Angled Views)
 - GPT-4 struggles with images of the board that are not viewed from a top-down perspective.
 - Inaccurate analysis when the board image is captured at an angle.
- Playing the Game (GPT 4 Turbo vs. GPT 4o)
 - GPT 4o is very bad at playing Tic Tac Toe. Rarely makes good or valid moves when asked for the next move.
 - GPT 4 Turbo plays the game better and makes more valid moves. Occasionally, when there are many pieces on the board, GPT 4 Turbo may lie about the board state to place a piece to win. This behavior, while interesting, indicates an inconsistency in game play.

Future Improvements

- Enhancing Physical Interaction (Playing on a Physical Board)
 - Code the robot to interact with and move physical pieces
- Computer Vision Enhancements
 - Implement cv2 algorithms to correct the perspective of the captured image (homography)
- Improving Game Play (Optimizing Prompts)
 - Experiment with different prompts to ensure Cozmo makes valid and strategic moves
 - Achieve consistent, high-quality decision-making in gameplay
- Expanding Game Complexity (Larger Board Sizes)
 - Expand the game to include 4x4 or 5x5 boards
 - Increase the difficulty and complexity of the game

Demo Video

