Playing Tic Tac Toe with Cozmo!
By Kanvi Shah
Project Description:
Using the cozmo python SDK, my project will provide an interactive and physical environment for a human user to play tic tac toe with a robot. The game will occur on paper, with the board drawn in front of the players. In order to make a move, Cozmo will move to the spot on the board that he wishes to place his X or O and display this on his screen. The user will be able to use a marker to make the physical moves.

Competitive Analysis:
Cozmo drawing: https://www.youtube.com/watch?v=Rl7JLXVaCBw
This was a demo of the Cozmo drawing SDK that was released. While it has nothing to do with tic-tac-toe per say, I will be able to make use of how Cozmo is able to draw letters (like X and O). This project uses a whiteboard which is different from my plan of using paper - however, I may procure a whiteboard if it turns out that is an easier surface for Cozmo to roll on.
Voice controlled tic-tac-toe with Cozmo: http://mdeng.me/portfolio/Cozmo/
This project was done by students in 15-494, where they taught Cozmo to recognize the game board and its different positions using codes like “alpha frog” for better voice recognition. Each player would tell Cozmo where to move their piece and the robot would move the corresponding color to that spot. My game will be different in that it will be user vs. robot instead of user vs. user. Also, I am trying to the original X’s and O’s instead of different colors, and instead of actual pieces, I am trying to actually have the game drawn onto a surface. However, I will need similar board detection and awareness of game logic.

Structural Plan:
The first section of functions will be used to detect and record the game board that Cozmo will be viewing with his camera. First, in the computer vision methods, he will have to pixelate his camera input, approximate the colors that he sees, and store this in a matrix. Next, I will have analysis functions that look for lines that resemble a tic tac toe board within this matrix. Here I will also have methods that recognize X’s or O’s already in the board and store this information.
The second section will contain the game playing algorithms that will help Cozmo use the board information and decide what move to make next. It will also contain switching player and game end methods, used to announce winners and ask for a new game to be started.
The third section will be all about Cozmo’s movements. Since he will have to in some way interact with the board to make moves, this is where the code for how to will reside. He will have methods to drive to a certain space and draw his piece there. Because I would like Cozmo to use a real marker, here will be the functions to write an X and an O. If the drawing does not work out, I will have backup methods here to display Cozmo’s piece on his screen instead.

Algorithmic Plan:

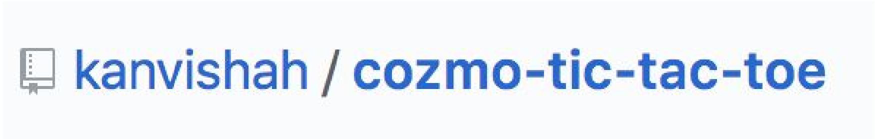
The most algorithmically complex part of my project will be to teach Cozmo to play tic-tac-toe. Currently, I plan on using the Minimax game theory algorithm. As it is well-documented, I will research and learn about how to use the algorithm to play tic-tac-toe. What I understand right now is that I will have to right a backtracking algorithm, that essentially assigns a value to every board state possible and while one side is trying to the minimum score, thee other side will be trying to get the maximum score. I will be breaking each part of the algorithm up, and only writing code as I understand.

Timeline Plan:

In the next few days, I would like all of Cozmo's computer vision worked out so he can recognize boards and moves madee on the board. By Sunday, I would like to have worked out the algorithmic game playing. By the TP2 deadline, next Wednesday, I would like to have Cozmo moving to the places he would like to play and displaying his move on his screen.

Version Control Plan:

I will be uploading my code to github, saving different versions as I add different features/change algorithms. My computer is also connected to my Google drive, so I will have version backup that way.



kanvishah / cozmo-tic-tac-toe

Module List:

I will be using the Cozmo robot as well as the corresponding SDK (cozmo) with it.

TP2 Update:

For my MVP, instead of using his camera to detect a board, Cozmo draws the board that he and human will play on. Currently, human move information is inputted through the computer. Cozmo draws both his moves and the human moves.

TP1 code for board detection did not work for game purposes. However, board detection and understanding is still in the plan for TP3. TP2 was made from scratch, encompassing around 300 lines of code, using structure found online of many other Cozmo robot projects.

In my research, I found another useful project that was used as a resource - two Cozmos play Horseshoe against a human:

https://github.com/cozplay/cozplay-demos/blob/master/horseshoe/horse_shoe.py

Similarly, I created a TicTacToe class that, when instantiated, begins a game of Tic Tac Toe. My game algorithms and physical game playing/drawing methods are all within this class. I plan to also include the board detection/understanding functions here as well. For TP3, I may make easy/medium/hard levels to increase complexity or add another two player game.

TP3 Update:

Cozmo now uses his camera to find the board and where to play. He makes both his moves and draws the human's moves.

Additional features include easy, medium, hard menu at beginning of game as well as a GUI for the human to use to play.

Each level corresponds to a different artificial intelligence algorithm - easy being random, medium being minimax, and hard being a "unbeatable" algorithm that finds forks for Cozmo to play in.