CAL POLY POMONA

SPOT PLAYS TIC-TAC-TOE

Ali El Lahib, Mt. San Antonio College

Principal Investigator: Dr. Daisy Tang

Graduate Mentor: Gabriel Alfredo M. Siguenza

Student Researcher: Ali El Lahib

Department of Computer Science

Cal Poly Pomona

California, United States

ABSTRACT:

This research investigates human-robot interactions through the attempted implementation of Tic-Tac-Toe gameplay using Boston Dynamics' robot Spot. Utilizing Python and Boston Dynamics' SDK, we programmed Spot to autonomously engage in Tic-Tac-Toe, with a focus on improving the precision of its piece placement and understanding of the game. A two-step approach was employed for piece placement, allowing Spot to accurately position game pieces without disturbing others on the board. However, the research encountered significant challenges with Spot's ability to consistently reorient itself towards the board and camera calibration issues at the intersection of the two front fisheye cameras, which have hindered the completion of a full Tic-Tac-Toe game with Sp. These limitations highlight the complexities of human-robot interaction and the need for further refinement of Spot's orientation and decision-making capabilities. Future work will involve addressing these issues and analyzing recorded interactions with Spot to further understand the human-robot interaction experience in game-playing social settings.

INTRODUCTION:

Human-robot interactions have become a critical area of study as robots increasingly participate in social and collaborative tasks alongside humans. Advanced robots like Boston Dynamics' Spot, with their mobility and manipulation capabilities, are well-suited for exploring interactive roles in structured activities, such as playing games. Recent advancements in computing power and artificial intelligence have further accelerated the development of humanoid robots by companies like Tesla and Figure, pointing towards a future where robots are more deeply integrated into everyday life. As robots become more prevalent, it is essential to investigate human-robot interactions to better understand these integrations, particularly in scenarios where robots must interact and make decisions in real time alongside humans.

This research aims to explore human-robot interactions by enabling game-playing abilities in Boston Dynamics' Spot, specifically through the implementation of Tic-Tac-Toe gameplay. Building on the foundational work of a previous group that developed the recognition and pickup of 3D-printed X pieces by Spot, this study focuses on expanding these capabilities. Key objectives include enabling Spot to autonomously reorient itself to the Tic-Tac-Toe board, accurately recognize and understand different stages of the game, and safely place its pieces. The primary goals are to improve piece placement precision, enhance game logic, and analyze the dynamics of human-robot interaction during gameplay.

Understanding these interactions not only advances the field of robotics but also provides insights into how robots can participate in structured, real-time activities with humans, potentially informing future applications in collaborative environments. The following sections will outline the materials and methods used, present the results, discuss the challenges encountered, and conclude with the remaining challenges and potential pathways for future research of this project.

MATERIALS:

- Robot: Boston Dynamics' Spot equipped with the Spot Arm.
- Game Pieces: 3D-printed X and O pieces for Tic-Tac-Toe gameplay.
- Tic-Tac-Toe Board: Consists of 9 fiducials, which are specially designed images similar to QR codes that Spot can recognize for board state identification. An additional 10th fiducial serves as a reference point for aligning Spot to the board.
- Python and Boston Dynamics Software Development Kit (SDK) for programming Spot, with a TensorFlow model for recognizing the X piece.

METHODS:

Board Setup

The Tic-Tac-Toe board was configured with 9 fiducials representing each space on the board grid, and a 10th fiducial serving as a reference point to help Spot maintain orientation relative to the board. The fiducials are held up on a wall by magnets.

Game Logic Implementation

For simplification, it is assumed that the player goes first with their O piece, and turns alternate with Spot placing its X piece next. Spot identifies where pieces are placed on the board by detecting which fiducials are obscured, assuming the piece is an O if it’s the player’s turn or an X if it’s Spot’s turn.

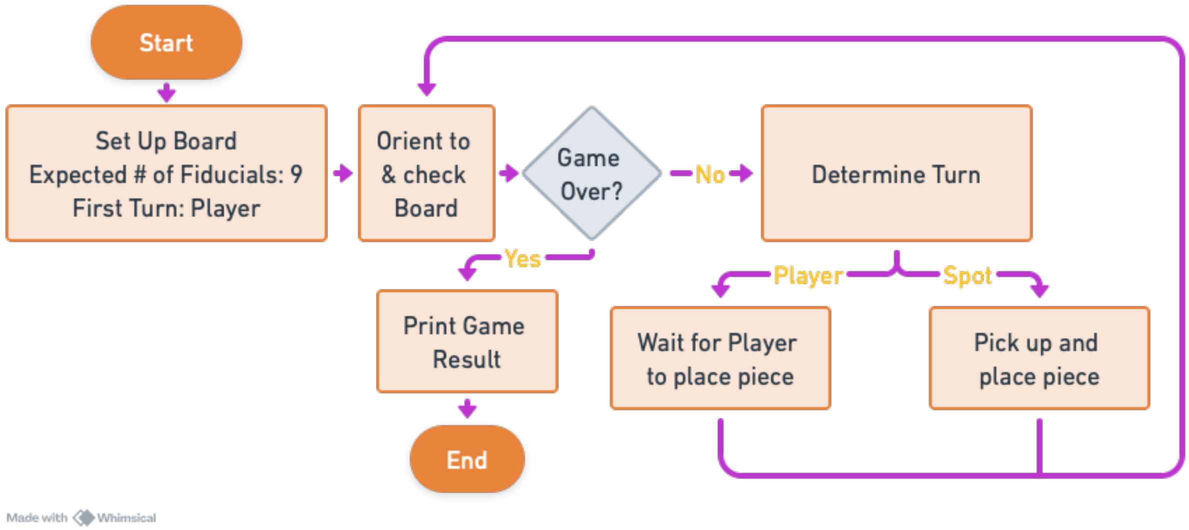

Figure 1: Logic flowchart of Tic-Tac-Toe gameplay for Boston Dynamics' Spot, showing decision-making for game stages, turns, and piece placement.

Figure 1 shows the logic flowchart for Tic-Tac-Toe gameplay between Spot and a player. Initially, the board is set up, and Spot is provided with what and how many fiducial IDs are there. It is assumed that the player places their piece first. Spot orients to the board, identifies the missing fiducial, and concludes that it has been covered by the player's piece. The program checks if the game is over; if not, it alternates turns. Spot then recognizes, picks up, and places its piece on the desired spot. The process repeats with Spot reorienting and updating the board state until the game concludes, with the result displayed on the computer screen.

Place Piece Implementation

A two-step approach was developed to enable Spot to place its piece on the board accurately without knocking other pieces off the board. In the first step, the arm takes a trajectory with an offset in the X direction (moving in and out of the fiducial) to align the piece-holding arm to the desired spot on the board. In the second step, the offset is removed, and the arm moves straight forward to place the piece precisely on the target spot.

To calculate the positions and trajectories, a Python script from the Boston Dynamics SDK called fiducial_follow.py was modified. Originally, this script guided Spot's body towards the first fiducial it detected. This was modified so that a desired fiducial can be specified and that Spot's arm would take the trajectory rather than its body.

Orienting to the Board

To orient Spot to the board, two measures are used. First, when Spot is booted up, it is positioned optimally facing the board, and these initial coordinates are recorded so Spot can return to them as needed. The second measure involves using the fiducial_follow.py script from the Boston Dynamics SDK, which originally directs Spot's body towards the first fiducial it detects. I modified the script to allow for a specified fiducial, which in this case uses the board's reference fiducial as the target for Spot's orientation. By using both these measures and adjusting the offsets appropriately, Spot can reliably maintain an optimal position and orientation where it can see all the necessary fiducials while also being within arm's reach to place its piece accurately.

RESULTS:

The implemented game logic in Python and using the Boston Dynamics SDK, enabled Spot to recognize the board state, reorient to the board, and autonomously recognize, pick up, and place its piece based on the assumed game flow. However, certain inaccuracies and issues occasionally disrupted the gameplay chain. For example, Spot sometimes failed to recognize specific fiducials due to poorly calibrated fisheye cameras. Additionally, there were instances where Spot incorrectly assumed it had picked up the piece when it had not. Although it is capable of recognizing a failed grasp and trying again as a result, this detection was not always consistent.

Figure 2: Spot with its piece placed accurately on the Tic-Tac-Toe board, using the two-step placement approach.

Figure 2 shows Spot successfully placing its X piece using the two-step placement approach described in the Methods section. When starting the project, the current placement piece function was not just very unsuccessful in placing the piece onto the board, but it took a trajectory that was prone to knocking other pieces off the board. With the two-step approach, Spot now is much more accurate and safer in placing its piece onto the board. Note, adjusting the offsets here could possibly improve the trajectory.

Utilizing the initial coordinates and the board reference fiducial proved to be very reliable in getting Spot to align to the board. However, there are instances where Spot is just a bit too far from the board making the placement piece function fail due to simply being out of arm's reach.

CONCLUSION:

This research successfully enabled Spot to place pieces on the Tic-Tac-Toe board safely and accurately, completed the game logic for Spot to understand and play Tic-Tac-Toe, and developed methods for Spot to reorient to the board. However, inconsistencies in Spot's positioning occasionally caused it to be too far from the board to place pieces correctly, preventing a fully recorded and analyzed game experience between a human player and Spot. Future work could focus on enhancing Spot's orientation capabilities, such as by adjusting offsets for better positioning, and exploring ways to eliminate the need for fiducials to make the setup more versatile. Additionally, expanding Spot's abilities to recognize, pick up, and place O pieces, as well as improving its existing interactions with X pieces, would further refine the gameplay.

While the primary goal was to explore human-robot interactions through Tic-Tac-Toe gameplay, the abilities developed, such as autonomous recognition, pickup, and placement of 3D-printed pieces, can have applications that extend beyond Tic-Tac-Toe. These capabilities can be adapted for other games and interactions, presenting other pathways for researching human-robot interactions. Also, the challenges encountered in implementing these abilities and those that remain show some of the complexities of developing robots that interact and make decisions in real time alongside humans. Addressing aspects like obstacle avoidance, speed adjustments, and handling unexpected situations will be crucial for advancing this field and improving our understanding of human-robot interactions.

ACKNOWLEDGEMENT:

I would like to express my sincere gratitude to my PI, Dr. Daisy Tang, for providing me the opportunity to work in her lab, offering valuable guidance on the various ways to proceed and think about the project, and supporting our weekly progress. Her insights helped me appreciate the project's broader impact.

I am also thankful to my graduate mentor, Gabriel Alfredo M. Siguenza, for teaching us the proper handling of Spot and guiding us through the project. Special thanks to Vinh Pham and his group for their foundational work on the project. Vinh helped in familiarizing us with their progress and the challenges left.

I am grateful to my lab partners, Sophia Quach and Deyi Chen, for collaborating with me on this project. Their efforts in resolving dependency issues and enhancing the model for recognizing the X pieces were helpful. They were a big help in putting together our research presentations, the support and teamwork is appreciated.

Thank you to Dr. Iraj Nejad, Dr. Alvin Kung, and Dr. Said Shokair for organizing the Mt. SAC Summer Research Program and providing me with my first research experience. The mentoring and support throughout the program were very helpful during the research experience.

Dr. Shokair's motivational speeches were inspiring and significant in shaping my approach to research and life goals.

Lastly, I want to extend my gratitude to the National Science Foundation for their financial support! Their award #2024276 helped reduce my financial stress and allowed me to focus significantly on this research opportunity.

REFERENCES:

Boston-Dynamics. (n.d.). Spot-SDK/python/examples/fiducial_follow at master · Boston-Dynamics/spot-SDK. GitHub. https://github.com/boston-dynamics/spot-sdk/tree/master/python/examples/fiducial_follow

F. Tang, V. Pham, R. Napinas, and G. A. M. Siguenza, “Research Experience for Undergraduates to Explore Human-Robot Interaction and Machine Learning with SPOT,” 2024.

Spot SDK. Spot SDK - Spot 4.0.3 documentation. (n.d.). https://dev.bostondynamics.com/