# Tom’s Target: an interactive cat-and-mouse game


1st Shiwen(Lareina) Yang
MEng in Computer Science
Cornell Tech
sy796@cornell.edu

2nd Qingxuan Yan
Connective Media
Cornell Tech
qy264@cornell.edu

3rd Xiang Chang
Connective Media
Cornell Tech
xc529@cornell.edu



***Abstract*—This paper presents TomTarget, an interactive robotic game inspired by a ’cat-and-mouse’ scenario, where the robot assumes the role of a cat attempting to catch players acting as mice. The game is designed to encourage dynamic human-robot interactions in a playful setting, drawing inspiration from musical chairs. Participants move around a designated area while the robot rotates and plays music. When the music stops, players must freeze, and the robot identifies and approaches any player in its field of view, delivering vocal cues to indicate game states.**

**The system employs a hybrid control architecture that combines reactive sensing and deliberative behavior planning. Reactive RGB camera data, processed through YOLO-based person detection, enables real-time player identification and tracking, while state-based planning ensures smooth, rule-based interactions. This design strikes a balance between responsiveness and structured gameplay.**

**To evaluate the robot’s social attributes, a user study was conducted with 17 participants using the Robotic Social Attributes Scale (RoSAS). The scale measured participants’ perceptions of the robot across dimensions such as interactivity, responsiveness, and negative attributes. Results indicated high scores in interactivity (4.7/7) and low scores in awfulness (1.75/7), demonstrating strong user engagement and minimal negative perceptions.**

**These findings highlight the effectiveness of simple, rule-based robotic behaviors in fostering enjoyable human-robot interactions. The study suggests that gamified robotic designs can enhance the social acceptance and perceived value of robots, particularly in interactive entertainment and educational settings, advancing the role of robots in Human-Robot Interaction (HRI) scenarios.**


Fig. 1. The TomTarget robot featuring a 3D-printed cat-themed design, including (a) the main body structure, (b) integrated RGB camera system, and (c) audio output components

## I. Introduction

Interactive robotic games present promising opportunities for technology education through engaging experiences. This paper presents TomTarget, an interactive robotic game designed to enhance K-12 students’ understanding of robotics technology while fostering their interest in STEM through playful interactions. The game implements a ”cat-and-mouse” scenario where players must avoid being caught by the robot within a constrained play area. Drawing inspiration from the classic cartoon ”Tom and Jerry”, the game features a cat-themed robot design implemented through 3D printing, with behavior patterns mimicking the pursuit characteristics of the cartoon character Tom. The gameplay mechanics uniquely combine this cat-and-mouse chase with elements adapted from musical chairs. This combination of familiar elements achieves both high accessibility for young participants and meaningful educational value.

The robot’s interaction system consists of two main components: movement and audio feedback. For movement, the robot rotates to search for players while playing music, and moves forward to catch detected players when the music stops. The audio system serves multiple purposes: providing game instructions (e.g., ”Game starts now!”), announcing game states (e.g., ”Got you!”), and expressing emotional responses to enhance engagement. These interactions guide players through the game while creating an entertaining experience.

The system implements a hybrid control architecture that combines real-time sensing with planned behavior execution through a finite state machine. The robot processes visual information from an RGB camera using YOLO-based person detection for player tracking. This architecture particularly suits the game’s requirements by enabling rapid response to player positions while maintaining structured gameplay sequences. The deliberative layer extends the system’s capabilities through an adaptive behavior module, implemented as a challenging ”cheat mode” during final rounds, demonstrating the architecture’s flexibility in supporting varied interaction patterns.

Field evaluations were conducted with twenty K-12 participants organized into four groups of five players each. The evaluation protocol consisted of three phases: (1) an introductory phase where participants received explanations of game mechanics and rules, followed by the robot's self-introduction to establish context; (2) gameplay sessions comprising five gameplay rounds lasting approximately five minutes; and (3) a post-interaction assessment phase where participants completed the Robotic Social Attributes Scale (RoSAS) questionnaire. Throughout the gameplay sessions, the participants showed active engagement throughout all sessions and developed diverse strategies to avoid capture.

## II. Method

### A. Game Design

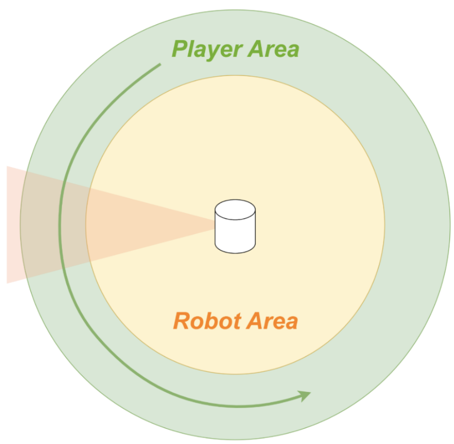


Fig. 2. Top-down view of the game layout illustrating the Robot Area, Player Area, and robot's detection range.

Our game implements an interactive "cat and mouse" scenario with clearly defined rules and mechanics. The game area is divided into two distinct zones: a circular "Robot Area" in the center (diameter: 2.3m) and a surrounding "Player Area" (outer diameter: 3m). The robot, positioned in the center, takes on the role of the cat (inspired by Tom from Tom and Jerry), while players act as mice in the outer ring.

Each round follows a structured sequence where the robot plays background music while rotating at a constant speed (0.4 rad/s) for a random duration between 12 to 22 seconds. During this phase, players must continuously move clockwise around the robot, keeping pace with the music. When the music stops, players must immediately freeze in their positions, and the robot simultaneously stops its rotation. If a player is detected within the robot's field of view (±30° from center), the robot initiates a capture sequence by moving forward and announcing "Caught you!" through its audio system. Caught players must exit the game area, reducing the active player count.

The players engage through strategic movement and positioning while maintaining continuous motion during the music phase. The game proceeds through five rounds, with victory achieved by either the last surviving player or the robot if it captures all players. To increase challenge and engagement, the robot enters a "cheat mode" in the final two rounds, where it stops immediately upon detecting a player rather than at random intervals, creating a dynamic interaction where players must carefully balance movement with strategic positioning.

### B. Finite State Machine (FSM)

*1) States and Behaviors:*

- **START:** Initialize motors, sensors, and audio system.
- **ROTATE:** Play music while rotating at a constant speed (0.4 rad/s) for 12-22 seconds.
- **STOP:** Stop rotation and music playback immediately.
- **ADJUST:** Align robot orientation by rotating until detected player is centered in camera view.
- **APPROACH:** Move forward at 0.2 m/s toward detected player.
- **CATCH:** Stop movement and play "Caught you!" audio.
- **BACK:** Return to origin through 180° rotation and forward movement.
- **END:** Conclude game and announce final results.

*2) State Transitions:*

- **START → ROTATE:** System initialization complete
- **ROTATE → STOP:** Timer expiry (12-22s) or player detection in final rounds (cheat mode)
- **STOP → ROTATE**: No player detected in the field of view.
- **STOP → ADJUST:** Player detected in field of view
- **ADJUST → APPROACH:** Player centered in camera view (±5 pixels tolerance)
- **APPROACH → CATCH:** Forward movement duration reached (3.0s)
- **CATCH → BACK:** Audio notification played three times, or the caught player left.
- **BACK → ROTATE:** Return to origin position complete
- **ROTATE → END:** Five rounds completed

*3) Sensor Data and State Transitions:* The robot's state transitions rely primarily on data from its onboard RGB camera, processed through YOLO-based person detection. This vision system provides continuous input for player tracking and positioning, with a detection range of ±30° from the robot's center line. The system processes this visual data to determine player presence and precise position (within ±5 pixels tolerance from image center). Additional onboard systems include timers for movement duration control and audio completion verification, as well as internal counters for tracking game rounds and audio repetitions.

*4) State Transition Table:* As shown in Table I

### C. Robot Control Architecture

Our system implements a hybrid control architecture, which is essential for our highly interactive game scenario. The robot's behavior must continuously adapt to player presence and movements, requiring both deliberative planning for game strategy and reactive responses for player detection. This hybrid approach allows us to implement two distinct game modes: a standard mode where the robot stops randomly

| State | Input | Output | Next | Condition |
|---|---|---|---|---|
| START | RGB image | Intro audio | ROTATE | Init complete |
| ROTATE | RGB + Timer | Angular vel: 0.4 rad/s, Music | STOP | Timer (12–22s) or detect in last rounds |
| STOP | RGB + Detection | Stop audio, Vel = 0 | ROTATE | No player detected |
| STOP | RGB + Detection | Stop audio, Magic audio, Vel = 0 | ADJUST | Player detected |
| ADJUST | RGB pos | Angular vel: ±0.4 rad/s | APPROACH | Center ±5px |
| APPROACH | Timer + RGB | Linear vel: 0.2 m/s | CATCH | 3.0s complete |
| CATCH | Detection + Count | Vel = 0, Catch audio | BACK | Audio ×3 |
| BACK | Timer + Pos | Phase1: Rot 180°, Phase2: Fwd 0.2 m/s | ROTATE | Return complete |
| ROTATE | Round count | End audio | END | 5 rounds done |

TABLE I
STATE TRANSITION TABLE WITH INPUTS, OUTPUTS, AND CONDITIONS.

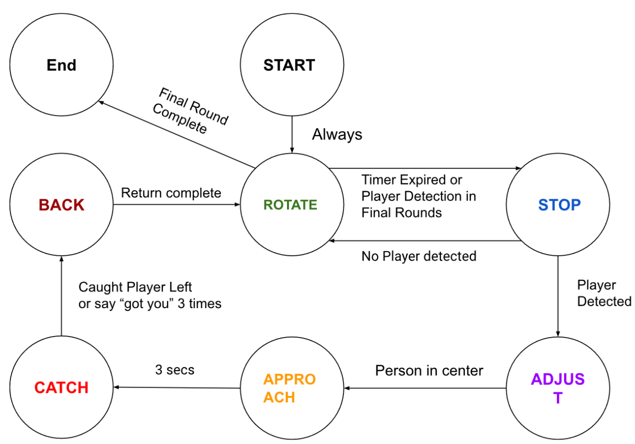


Fig. 3. FSM diagram showing game states, transitions, and their triggering conditions.

regardless of player positions, and a "cheat mode" in final rounds where the reactive layer directly triggers state transitions upon player detection. The architecture's flexibility enables such dynamic behavior adaptation while maintaining consistent game rules and player engagement.

*1) Implementation Details:* The architecture is implemented as a ROS2 node named `robot_catch_pkg` with the following components:

- **Subscribers:**
  - RGB camera image (`/color/image`)
  - YOLO-based person detection (`/color/mobilenet_detections`)
- **Publishers:**
  - Velocity commands (`cmd_vel`)
- **Parameters:**
  - Linear velocity: 0.2 m/s
  - Angular velocity: 0.4 rad/s
  - Center tolerance: ±5 pixels
  - Movement duration: 3.0s
- **Control Timer:**
  - 20Hz control cycle (0.05s) for state management and motion control

*2) Testing and Validation:* The system implementation was tested using various visualization and debugging tools. For the vision system, we employed OpenCV's visualization capabilities (`cv2.imshow`) to verify the RGB camera input quality and field of view. Person detection accuracy was validated by drawing bounding boxes around detected individuals in real-time, with visual indicators for detection confidence and position relative to the robot's center line. State transitions and motion control were verified through ROS2's logging system, which provided real-time feedback on state changes, timing parameters, and movement commands.

*3) System Components:* The system operates with both parallel and sequential processing components:

- **Parallel Components:**
  - Continuous RGB image processing and person detection
  - Audio system for music playback and game announcements
  - Timer monitoring for state transitions
- **Sequential Components:**
  - State-based behavior execution
  - Movement sequences (rotation, approach, return)
  - Audio notification sequences

### *D. Robot Control Architecture*

**Chosen Architecture and Justification:** We implemented a **hybrid control architecture**, combining reactive sensing and deliberative behavior planning to ensure both real-time responsiveness and structured gameplay. The reactive layer uses RGB camera data processed via YOLO-based person detection for real-time player tracking, while the deliberative layer employs a Finite State Machine (FSM) to orchestrate high-level behaviors such as music-based movement, player identification, and vocal feedback. This hybrid approach supports dynamic adaptation to player presence, enabling two distinct game modes: a standard mode with randomized stops and a "cheat mode" in the final rounds, where the reactive layer triggers immediate state transitions upon player detection.

**Implementation Details:** The architecture is implemented as a ROS2 node package named `robot_catch_pkg`, with the following components:

- **Subscribers:** - RGB camera image (`/color/image`) for vision input. - YOLO-based person detection results (`/color/mobilenet_detections`).
- **Publishers:** - Velocity commands (`cmd_vel`) to control robot movement.
- **Parameters:** - Linear velocity: 0.2 m/s, Angular velocity: 0.4 rad/s. - Center tolerance: ±5 pixels for player alignment. - Movement duration: 3.0 seconds, Control timer frequency: 20Hz (0.05s).
- **State Transitions:** The FSM handles sequential states: `START` → `ROTATE` → `STOP` → `ADJUST` → `APPROACH` → `CATCH` → `BACK`. These transitions depend on timers, sensor input, and player detection results.
- **Timers:** - Control cycle at 20Hz for state management and motion control. - Randomized rotation durations (12–22 seconds) with a minimum threshold for cheat mode.

**Testing and Validation:** We employed OpenCV's visualization (`cv2.imshow`) to verify RGB camera input quality and person detection accuracy by drawing bounding boxes in real-time. ROS2's logging tools were used to monitor state transitions, timing parameters, and velocity commands during testing on the Turtlebot4 platform.

**Parallel and Sequential Components:** The hybrid control architecture integrates both parallel and sequential components:

- **Parallel Components:** - Continuous RGB image processing and YOLO-based person detection. - Audio playback system for music and game announcements. - Timer monitoring for state transitions.
- **Sequential Components:** - FSM-driven state execution: Rotation (ROTATE), stopping (STOP), alignment (ADJUST), movement (APPROACH), and catching (CATCH). - Audio notifications synchronized with state transitions.

**Control Architecture Diagram:** Figure 4 illustrates the robot control architecture, depicting the integration of inputs (RGB data), processing (YOLO detection and FSM), and outputs (velocity commands and audio cues).

This system was rigorously tested within a constrained play area, demonstrating accurate player detection, smooth state transitions, and responsive audio feedback, which collectively ensured engaging gameplay.

### *E. Robot Design and Prototyping*

**Prototyping Techniques:** The robot design process involved an iterative approach using a combination of sketching, storyboarding, and physical prototyping.

- **Sketching:** Initial sketches were created to conceptualize the robot's cat-like appearance and its role in the game.

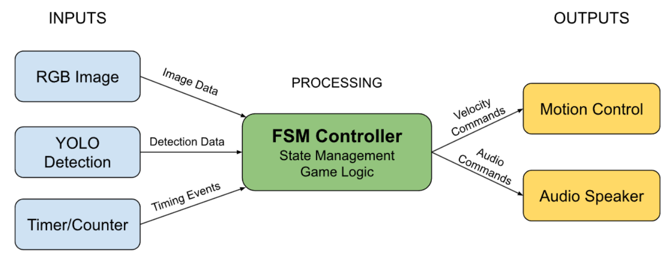


Fig. 4. Robot Control Architecture: Hybrid integration of reactive sensing and FSM-based behavior planning.

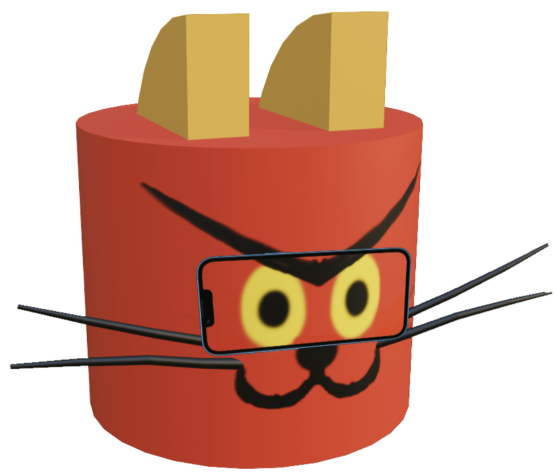

Fig. 5. Initial sketches and storyboard for robot design.

- **Cardboard Prototyping:** A low-fidelity prototype of the robot body was initially constructed using cardboard to test the size, movement feasibility, and visibility of onboard sensors like the RGB camera.
- **3D Printing:** High-fidelity prototyping was achieved using 3D printing to fabricate durable, precise components for the robot's body.

**Figures as Evidence:** Figures below showcase various stages of the prototyping process, from initial sketches to final 3D-printed components:

**Choice of Materials and Fabrication Methods:** To balance durability, precision, and aesthetics, the following materials and fabrication methods were chosen:

- **Laser Cutting:** Laser-cut acrylic sheets were used for the circular base and top of the robot body. Acrylic provides a clean, polished appearance and is lightweight while maintaining structural stability.
- **Cardboard (Initial Prototype):** Cardboard was used in the early stages to rapidly prototype and evaluate the robot's dimensions, movement feasibility, and camera visibility.
- **3D Printing:** PLA filament was used for the robot's body and cat-themed features due to its affordability, lightweight nature, and ease of fabrication. The 3D-

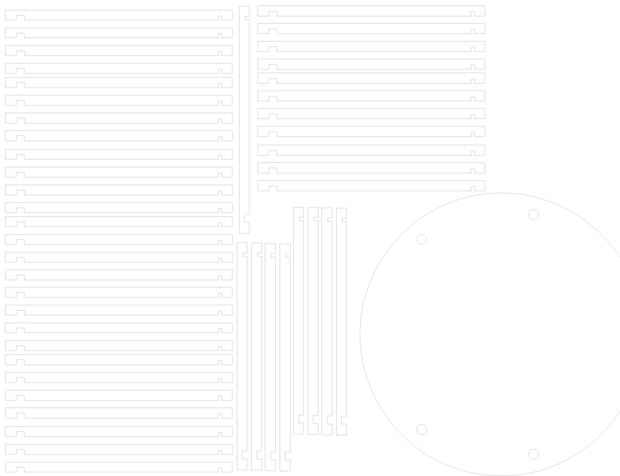

Fig. 6. Laser-cut acrylic components used for the robot's base and top structure.

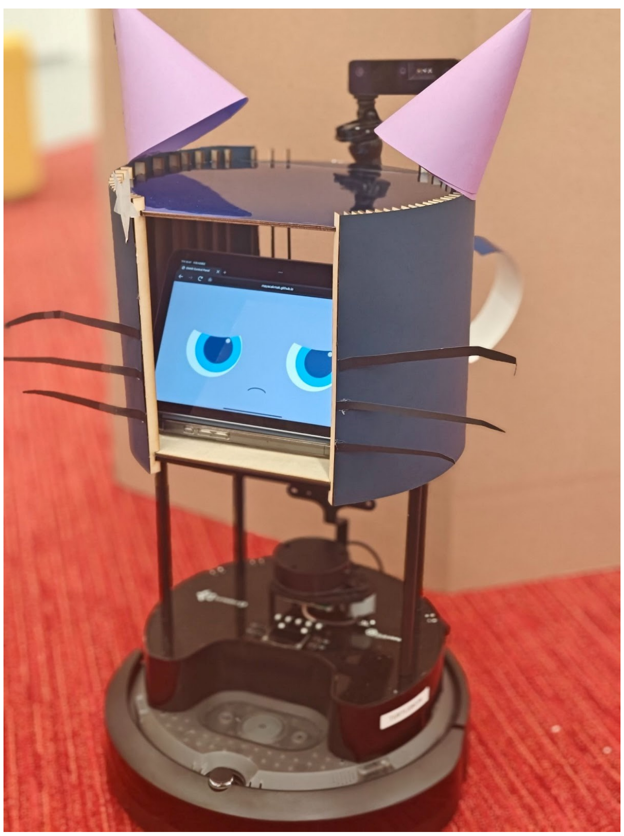

Fig. 7. Cardboard prototype used for testing body structure and sensor placement.

printed parts ensured structural integrity and accommodated openings for the camera and mounting points.

By combining low-fidelity cardboard prototyping with high-fidelity 3D printing and laser cutting, we successfully created a visually appealing and functional robot that aligns with the game's playful theme while supporting the required sensors and hardware components.

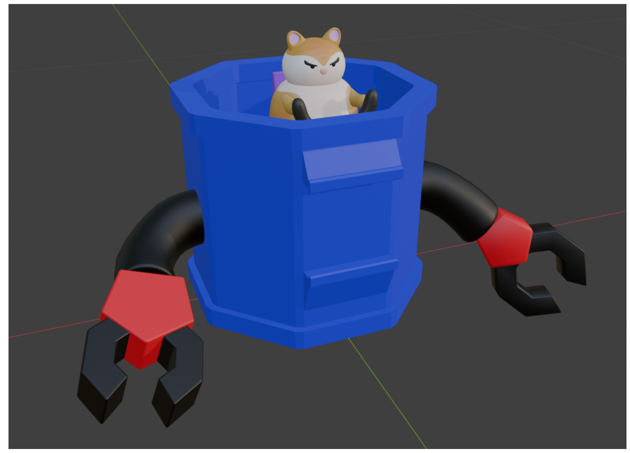

Fig. 8. 3D-printed components, including the body structure and cat-themed features.

## III. Procedure

### A. Participants

We recruited participants as part of a **public outreach event** organized under the **K-12 Initiative at Cornell Tech**. The recruitment process targeted students from the Dock Street School for STEAM, where participants engaged in the interactive robotic game during scheduled rotations. This event provided an excellent opportunity to test the robot with diverse participants in a controlled, educational environment.

A total of **17 participants** were involved in the study, divided into smaller groups of students each to ensure organized gameplay and data collection.

The age range of participants was **10 to 14 years old**, as the event focused on K-12 students, specifically middle school attendees, to match the playful and engaging nature of the "TomTarget" game.

We employed a **convenience sampling** technique for participant recruitment. This method was suitable as the participants were already present and willing to engage in the activities scheduled for the K-12 outreach program.

### B. User Studies

**Study Procedure:** The study was conducted in three phases to ensure smooth execution and data collection:

- **Introduction (1-2 minutes):** Participants were welcomed to the study and introduced to the rules and mechanics of the "TomTarget" game. The research team explained that the robot, acting as a "cat," would attempt to catch players who must freeze when the music stops. Participants were informed of the study's safety measures, and consent was obtained.
- **Game Play (6 minutes):** Participants actively engaged in the game. The robot rotated, played music, and stopped at random intervals. During each stop, the robot detected players in its field of view and approached the nearest target to call out "Caught you!" This sequence continued for multiple rounds until the game concluded.

- **Administration of Self-Report Measures (1-2 minutes):** After the gameplay, participants completed the Robotic Social Attributes Scale (RoSAS) questionnaire. This survey assessed their perceptions of the robot's social attributes, such as responsiveness, friendliness, and interactivity.

**Game Environment Setup:** The game was conducted in a designated play area divided into two zones:

- **Robot Area:** The central zone where the robot operated, rotating and detecting participants.
- **Player Area:** A square area surrounding the robot, marked with tape to help participants maintain consistent distances while moving.

The environment setup included external speakers to play background music, enhancing participant engagement. The robot's RGB camera was calibrated before the study to ensure accurate player detection, and the fixed-radius play area allowed for precise control of robot movements. Pre-study preparations also involved verifying the robot's state transitions, sensor accuracy, and music playback synchronization.

**Study Testbed Figures:** Figures illustrating the study testbed, including the robot and environment setup, are provided below:

### C. Evaluation Metrics

The primary evaluation metric for this study was the **Robotic Social Attributes Scale (RoSAS)**, a validated self-report measure used to assess participants' perceptions of the robot's social attributes during the game. Participants completed the RoSAS questionnaire immediately after gameplay, ensuring that their responses reflected real-time impressions of the robot's behavior.

Participants rated 18 attributes on a **7-point Likert scale** ranging from 1 (*Not at all*) to 7 (*Very much so*). These attributes included:

- Dangerous, Awkward, Aggressive, Feeling, Strange, Knowledgeable, Reliable, Happy, Compassionate, Awful, Competent, Social, Responsive, Scary, Capable, Emotional, Interactive, Organic (Non-mechanical).

The collected RoSAS data provides quantitative insights into participants' subjective experiences of the robot, particularly focusing on its perceived social capabilities, friendliness, and any discomfort caused. This evaluation method was chosen for its ability to capture multidimensional feedback, which is essential for assessing the robot's effectiveness in a human-robot interaction context.

## IV. Results

The descriptive statistics for the self-report measures indicate the following:

- The attribute with the **highest average score** is **Interactive** with a mean score of **4.69**, indicating participants found the robot to be engaging and responsive during the interaction.

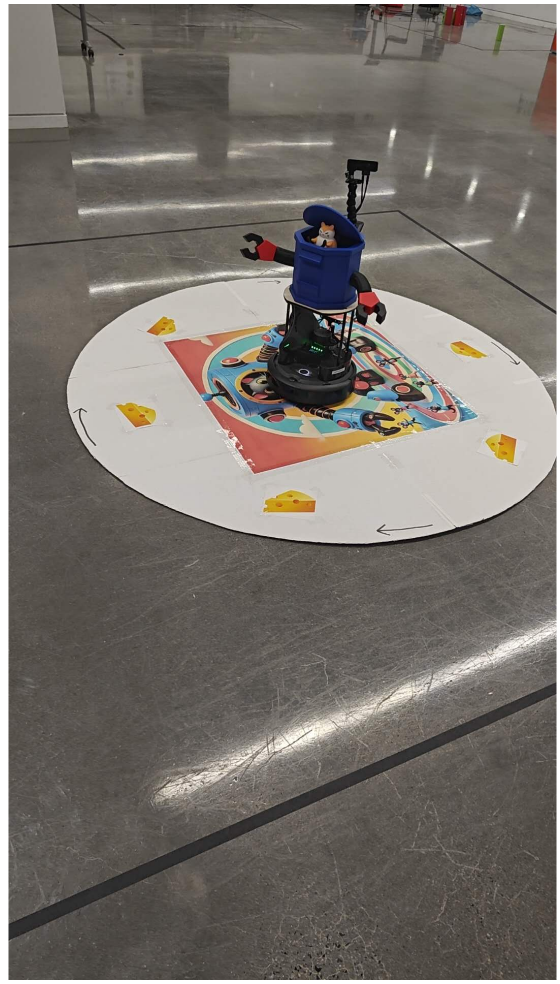

Fig. 9. Study testbed showing the robot in the Robot Area and participants in the Player Area.

- The attribute with the **lowest average score** is **Awful** with a mean score of **1.75**, suggesting that participants perceived minimal negative characteristics in the robot's behavior.

To provide a visual representation of the average scores, a bar chart is included below:
These results highlight the robot's strengths in interactivity and responsiveness while indicating minimal negative perceptions among participants.

## V. Conclusion

**Surprising Results:** One of the most surprising findings was the attribute **Interactive** receiving the highest average score of **4.69**. This result indicates that participants found the robot highly engaging and responsive, which exceeded our initial expectations given the relatively simple state transitions and limited expressiveness of the robot. Conversely, the attribute **Awful** had the lowest average score of **1.75**, suggesting that participants perceived minimal negative or uncomfortable aspects of the robot's behavior, which aligns with our goal of creating a smooth and enjoyable experience.

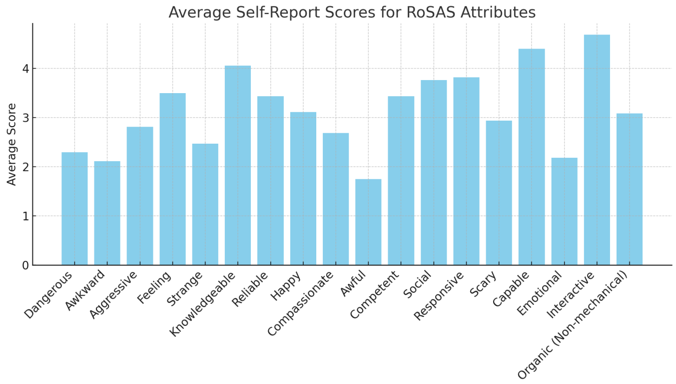


Fig. 10. Average Self-Report Scores for RoSAS Attributes

**Future Improvements:** In future iterations of the project, we would focus on enhancing the robot's social and expressive capabilities to further improve user engagement. Specific improvements include:

- Adding dynamic facial expressions and voice tones to reflect the robot's internal state.
- Implementing adaptive behavior using machine learning to allow the robot to respond in real time to participant actions and feedback.
- Conducting longer and more varied gameplay sessions to capture a broader range of user interactions and perceptions.

**Limitations:** Several limitations of the study may have influenced the results:

- **Sample Size:** Due to time constraints, only **17 participants** were able to participate in the study, which significantly limits the generalizability of the findings.
- **Questionnaire Quality:** Many of the completed questionnaires were of poor quality, with participants failing to answer all questions or providing inconsistent responses, which may have affected the reliability of the results.
- **Environment Interference:** During gameplay, the presence of other people in the surrounding environment occasionally interfered with the robot's target detection, leading to minor inaccuracies in detecting players. However, these issues did not affect the overall game progression.
- **Robot Behavior:** The robot's behavior was constrained to pre-programmed states, limiting its adaptability to complex or unexpected participant actions.
- **Implementation Constraints:** Hardware and sensor limitations, such as occasional detection delays or inaccuracies, affected the precision and timing of the robot's responses during gameplay.

**Future Work and Lessons Learned:** Based on the findings and limitations, the following directions are proposed for future work:

- Expand the participant pool to include a more diverse range of ages, backgrounds, and group dynamics to improve the generalizability of results.
- Ensure adequate time for recruiting participants and conducting studies to avoid limitations in sample size and data quality.
- Improve the robot's target detection algorithms to minimize the impact of environmental interferences, ensuring greater accuracy in detecting players.
- Integrate advanced AI algorithms, such as reinforcement learning or neural networks, to enable real-time adaptive behavior and enhance interactivity.
- Improve hardware and sensor integration to minimize latency and enhance the robot's perception accuracy during gameplay.

Through this project, we learned that even relatively simple robotic behaviors, when thoughtfully designed, can lead to positive user experiences. However, we also learned that sufficient time, preparation, and environmental considerations are critical for ensuring the quality and reliability of collected data. Balancing system robustness with interactivity remains critical for creating engaging human-robot interactions that are both reliable and enjoyable.

## VI. GitHub Code

Github URL:
https://github.com/crazycatseven/HRI-Tom-Target.git

## VII. Video Demonstration

The video demonstration can be accessed at the following link:

Link Here