Research Article/Generic Engineering

http://doi.org/10.15446/ing.investig.108904

An Intelligent System-Based Strategic Plan for a Humanoid Robot Playing the Game of Dominoes

Plan estratégico basado en sistemas inteligentes para el juego de dominó por parte de un robot humanoide

Álex Medina[1], Daniela Charris[2], Mauricio Pardo[3], and Christian G. Quintero M.[4]

ABSTRACT

The application of intelligent systems in humanoid robots provides research and development alternatives, as is the case with human-robot interaction. This paper focuses on the design and implementation of an intelligent system in the NAO robot to plan and execute moves in the board game known as *dominoes*. This system uses the NAO robot's vision to determine the distribution of tiles on the board, as well as those available in hand. The appropriate moves are planned using a computational intelligence technique, and a kinematics model executes them. The results show that the vision system has an average error of 5.62%, in addition to 3.37% for the decision-making system and 7.87% for the kinematics of the robot. This leads to the NAO robot being capable of making successful plays through the proposed system, with an average effectiveness of 83.15%.

Keywords: computer vision, decision tree, dominoes, board games, forward kinematics, human robot interaction, image processing, intelligent system, NAO robot, robotics

RESUMEN

La aplicacion de sistemas inteligentes en robots humanoides brinda alternativas de investigacion y desarrollo, como es el caso de la interaccion humano-robot. Este trabajo se enfoca en el diseño e implementacion de un sistema inteligente en el robot NAO para planificar y ejecutar movimientos en el juego de mesa conocido como *domino*. Este sistema utiliza el sistema de vision del robot NAO para determinar la distribucion de fichas en el tablero y de las disponibles en la mano. Los movimientos adecuados se calculan mediante una tecnica de inteligencia computacional, y un modelo de cinematica los ejecuta. Los resultados muestran que el sistema de vision tiene un error promedio del 5.62 %, ası como del 3.37 % para el sistema de decision y de 7.87 % para la cinematica del robot. Esto lleva a que, a traves del sistema propuesto, el robot NAO sea capaz de realizar jugadas exitosas con una efectividad promedio del 83.15 %.

Palabras clave: vision por computadora, arbol de decision, domino, juegos de mesa, cinematica inversa, interaccion humano-robot, procesamiento de imagenes, sistema inteligente, robot NAO, robotica

Received: January _th 20xx
Accepted: January _th 20xx

Introduction

Humanoid robots' incorporation of intelligent systems creates new opportunities for research and development in fields such as human-robot interaction (HRI). HRI has recently been employed in a variety of sectors, including education (Barnes *et al.*, 2019; Budiharto, Cahyani, Rumondor, and Suhartono, 2017; Jeon *et al.*, 2017), sports (Bi, Yang, Zhang, Goupil, and Feng, 2018), entertainment (Knox and Watanabe, 2018), business and industry (Asfour *et al.*, 2018; Barakeh, Alkork, Karar, Said, and Beyrouthy, 2019), and healthcare (Ajani, Obasekore, Kang, and Rammohan, 2023; Mohan and Kuchenbecker, 2019; Schadenberg, 2019), among others.

For a friendly HRI, three fundamental components are necessary: environment detection (vision system), planning (decision system), and movement actions (execution system). In particular, the NAO robot's vision system has been used for environment exploration through object detection (Sun, Wang, Zheng, and Liu, 2023; Zhu, Yi, Chellali, and Feng, 2018), object tracking (Juang and Zhang, 2019), image segmentation (Yan, Li, Liu, Liu, and Chen, 2022), visual navigation (Lobos-Tsunekawa, Leiva, and Solar, 2018), and transfer learning (Ovalle-Magallanes *et al.*, 2021). Regarding its decision system, the NAO robot has been used for sentiment analysis (Goenaga, Navarro, Quintero, and Pardo, 2020; Díaz, Shaik, Santofimio, and Quintero, 2018) and to support psychologists during treatments (Cao *et al.*, 2020; Feidakis *et al.*, 2023; Yang *et al.*, 2019). As for the execution system, kinematic methods have been studied to perform specific tasks, such as imitating human arm movements (Wei, 2020; Zhai, Wen, Zhu, and Guo, 2017) and walking (Piperakis, Koskinopoulou, and Trahanias, 2018; Wang, Xue, and Chen, 2020), among others.

[1]Mechatronics Engineering, Universidad de las Fuerzas Armadas, Ecuador. MSc Electronics Engineering, Universidad del Norte, Colombia. Affiliation: Member of the Robotics and Intelligent Systems Research Group, Universidad del Norte, Colombia. E-mail: medinafa@uninorte.edu.co
[2]Electronics Engineering, Universidad del Norte, Colombia. MSc Electronics Engineering, Universidad del Norte, Colombia. Affiliation: Member of the Robotics and Intelligent Systems Research Group, Universidad del Norte, Colombia. E-mail: dmcharris@uninorte.edu.co
[3]Electronics Engineering, Universidad del Norte, Colombia. MSc and PhD in Electrical Engineering, Georgia Institute of Technology, USA. Affiliation: Assistant professor and member of the Robotics and Intelligent Systems Research Group, Universidad del Norte, Colombia. E-mail: mpardo@uninorte.edu.co
[4]Electronics Engineering, Universidad Industrial de Santander, Colombia. MSc and PhD in Information Technologies, Universidad de Girona, Spain. Affiliation: Associate professor and member of the Robotics and Intelligent Systems Research Group, Universidad del Norte, Colombia. E-mail: christianq@uninorte.edu.co

More specifically, robots playing board games have increasingly gained attention for their multiple benefits. Robots can play with people, enhancing interaction enjoyment and social engagement (Lin, Ng, and Sebo, 2022). However, humans may find it challenging to compete against robots in immersive entertainment settings. The cited study examines how players' gaming experiences vary depending on whether a human or an embodied robot serve as the game guide. In addition, people who do not compete with other humans can gain social experiences via robots, and students can learn programming and other STEM skills by playing board games with robots, as indicated by (Inoue, Jimenez, Haruta, and Oonuki, 2022), whose study evaluates how robots affect students' perceptions of learning and their motivation to learn.

As for teleoperation roles, robots can deliver haptic feedback in addition to visual information, which enhances the user experience (Moya, Slawiñski, Mut, and Wagner, 2021). Moreover, robotic board games can boost creative capacity (Mercier and Lubart, 2021). In particular, humanoid robots are used for research in different types of games, where they play against a person or are required to play for themselves. Several humanoid robots have been used to play chess by processing board and tile images (Kołosowski, Wolniakowski, and Miatliuk, 2020; Patil, Fegade, Kadam, Patil, and Singhaniya, 2021; Rath, Mahapatro, Nath, and Dash, 2019). In (Juang, 2022), a NAO robot was employed to this effect. Another NAO robot capable of playing Simon Says was presented in (Li, Imeokparia, Ketzner, and Tsahai, 2019).

Current advances show the potential of HRI in board games for a variety of purposes, including education, entertainment, and research. The goal of robots playing board games is to make the experience more engaging and challenging for humans. Recent research has focused on developing algorithms that can learn to play high-level board games. Machine learning techniques such as evolutionary computation, neural networks, and reinforcement learning can be used by robots to improve their gameplay over time (Karunanayake *et al.*, 2020). In particular, reinforcement learning techniques in robotics can be developed and tested using robots, enabling the system to complete tasks that would normally need a sophisticated decision-making algorithm (Karmanova, Serpiva, Perminov, Fedoseev, and Tsetserukou, 2021).

Robots can learn to play board games utilizing a variety of techniques, including programming instruction, machine learning, and cognitive human-robot interfaces (Hu, Zhao, Meng, and Wu, 2020). Yet, how well they work depends on the objectives and the study goals. In short, the effectiveness of robots in playing board games is determined by a variety of factors such as the complexity of the game, the robot's level of autonomy, and the quality of the algorithms used. Robots may have an advantage in terms of strategy and decision-making in certain scenarios, where they may use machine learning and reinforcement learning techniques to discover complex behaviors and strategies in real time. Humans may still have an advantage in situations where creativity, intuition, and social interaction are important decision-making factors. Robots may be incapable of adapting to unexpected moves or strategies employed by human opponents, and they may be unable to comprehend the social and emotional aspects of gameplay that humans value. However, by using intelligent techniques, robots can learn and improve their gameplay. In this sense, one of the main challenges in designing robots for board games is to ensure that the robot understands the game rules and can make appropriate decisions. This calls for the creation of algorithms capable of interpreting the game state and determining the best move to make. Another challenge is creating a way for the robot to interact with physical game pieces that is intuitive and natural for human players. This could entail creating specialized sensors or grippers that can handle various types of game pieces. Another challenge is to make the robot socially acceptable to human players. This includes considering aspects such as the robot's appearance, behavior, and level of autonomy. A robot that is overly competitive or aggressive, for example, may be off-putting to human players, whereas a robot that is overly passive may not provide a satisfying challenge. In this sense, the robot's design can influence how humans perceive and interact with it during gameplay. The ultimate challenge is to design the robot so that it can adapt to different types of board games, which calls for the development of algorithms and hardware capable of handling a wide range of game dynamics and rules.

To summarize, designing robots for board games involves a variety of challenges related to game understanding, physical interaction, social acceptability, and adaptability. Overcoming these challenges necessitates a multidisciplinary approach that combines expertise in robotics, artificial intelligence, HRI, and game design. For example, (Karunanayake *et al.*, 2020) describes a real-time vision-based robotic system that plays Carrom against a skilled human opponent, and (Raghavan, Srinivasan, Dey, and Chandar, 2021) presents an automated laser alignment and image processing-based robotic Carrom player. The above references demonstrate the potential for robots to play board games, but there are no widely known examples of successful robots designed specifically for this purpose. In this vein, this paper aims to contribute to the field of HRI entertainment, specifically board games. Dominoes was chosen for this study because it is a traditional game from Central America and the Caribbean. This study examines whether a humanoid robot is capable of interacting with people and planning and executing its plays based on game conditions (tiles on the table and those in hand) via an intelligent system-based strategic plan.

This paper is structured as follows. First, an explanation of the game of dominoes is presented in the *Materials and method* section. Afterwards, the proposed approach is divided into vision, decision, and execution systems, outlining each aspect. Then, the complete system is presented and demonstrated via a user interface. Finally, the *Results and discussion* section describes the tests carried out to validate the system's performance.

Materials and method

Dominoes board game

The most common game of dominoes consists of 28 tiles, each with two numbers represented by pips and blanks, as shown in Fig. 1. There are seven tiles of each number, going from zero to six. In the most popular game mode, four players participate in either pairs or score modes. In this paper, a three-player game variant is explored in order to implement individual game strategies, as the robot's vision of the available tiles in hand covers a maximum of nine.

Figure 1. Dominoes game
Source: Authors

For the three-player variant, the blank-blank tile is discarded, so that nine tiles correspond to each player. The player who draws the double-six tile starts the game. However, if it is a recurring game, the winner of the last set starts a new one with the double-six tile, but, if they do not have it, they can start playing with the highest double tile they have. If the player does not have a double tile, then the tile with the most pips comes out. The starting tile is placed face-up at the center of the board. The next player must play to match the blanks/pips on either side of the starting tile.

In the following rounds, the players take turns repeating the procedure. As the game unfolds, there will be two available ends to place a playing tile. However, if a player does not have a tile that matches the blanks/pips at either end, they pass the playing turn to the next player. The game ends when a player manages to place all their tiles or when the game is closed (no player can place a tile on the board). The player with the least pip count wins the set (Tres, 2014).

Game theory deals with the study of games like dominoes, classifying them as *deterministic* or *non-deterministic*. The former are those for which it is possible to generate a winning strategy. A sequence of steps or *strategy* is said to be winning because it increases the chance of winning for the player who applies it, regardless of the circumstances. Non-deterministic games, in turn, are those for which there is no winning strategy. In the case of dominoes, the presence of *randomness* implies that it is not possible to determine a strategy that always wins. For this research, we selected a non-deterministic game, since the aim is for the players to interact with the robot, winning or losing the games in which they participate.

Proposed approach

This paper describes an intelligent system as a strategic plan that allows a NAO robot to play the game of dominoes. The proposed system is divided into three subsystems: image processing (vision), computational intelligence (decision-making), and robot kinematics (execution). The components of each subsystem are presented in Fig. 2.

The image processing subsystem captures what the robot perceives of its surroundings with its built-in camera. Through this process, the most relevant variables of the game are obtained, such as the tiles on the table and those available in hand. These data are then entered into the computer system. In the computational intelligence subsystem, a decision tree-based algorithm is executed, so that the robot can decide on the tile to be played. This decision tree considers several game strategies collected from people with medium to high experience in the game. Finally, the robot kinematics subsystem provides the most viable way to execute a play, taking the selected tile from those available in hand and placing it correctly on the board. For this work, direct kinematics was proposed and then experimentally verified by a model that determined its viability, as it minimized the error when choosing a tile.

The vision system includes two cameras to separately observe the tiles available in hand and those on the board. The resulting data are then entered into the image processing block to obtain the game variables.

As observed in Fig. 2, there is an interconnection block, *i.e.*, game variables acquisition, that continues into the decision system. It was devised using a voice recognition algorithm, and its role is to let the NAO robot know when its turn to play has begun.

Once the game variables have been acquired, the decision system processes the tiles available on the table and the ones in hand, using an intelligent system that provides the final decision, *i.e.*, the tile to be played. Once said tile has been selected, the execution system takes the decision system output as input to the direct kinematics model of the NAO robot. The forward kinematics direct one of the robot's arms to pick the selected tile and place it on the board, which is followed by a voice command.

Finally, the main system returns to the game variables acquisition block, where the NAO robot waits for the voice command to resume the game.

Game scenario

For the proposed system, we decided that at least three players would play the game, as the NAO robot's field of vision can cover up to nine tiles, given the tile set selected for testing. Then, the tile and board colors were established. In this case, white tiles with black pips and a blackboard were selected, so that there would be a better contrast for the robot's vision and image processing. Similarly, the game area was delimited, defining the location of the tiles in hand for the NAO robot and the gaming space to place the tiles on the board. The tiles in hand had to be located in two planes (Fig. 3), so that the NAO robot could visualize them correctly and they were within reach of its arms.

Vision system

To process the images of the tiles in hand, an algorithm was developed, the partial results of which are shown in Fig. 4. As observed, the objective is to detect all the tiles together with their corresponding pips, leading to the identification of the associated number.

To this effect, the Edge Detection algorithm provides information about object boundaries, and it is employed for analysis and filtering applications. Its goal is to drastically reduce the amount of data to be processed while preserving structural information. This algorithm's working principle detects sudden color intensity changes and enhances them, highlighting the resulting image edges. It is very effective in

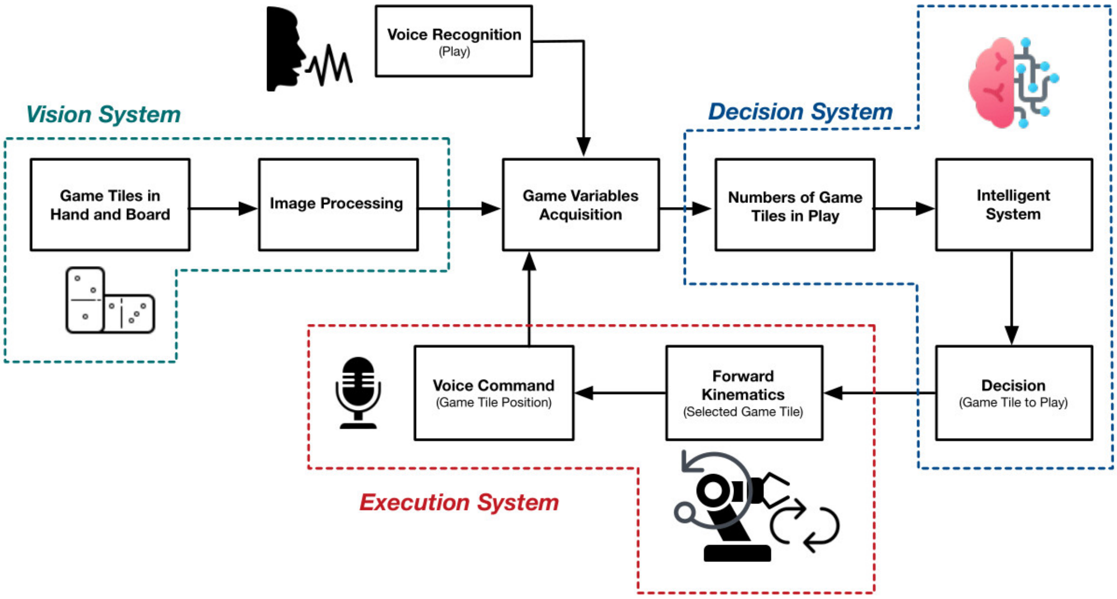

Figure 2. Diagram of the proposed approach
Source: Authors

Figure 3. Game scenario for the proposed system
Source: Authors

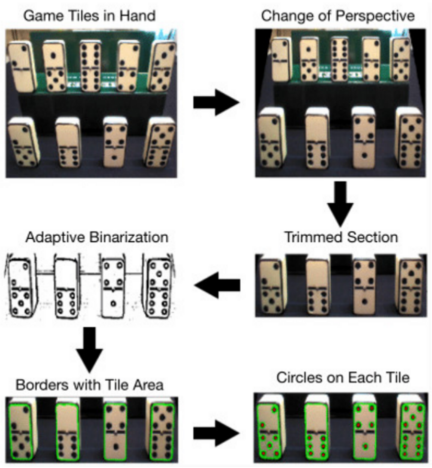

Figure 4. Algorithm for the detection of the game tiles in hand
Source: Authors

accentuating contrast and detecting isolated spots or small details.

One of the algorithms most commonly used in this field is Canny, which looks for the maximum gradient along an edge. The result is a border/non-border binary threshold-adjustable image, as shown in Fig. 4 (adaptive binarization). In addition, the Segmentation algorithm subdivides an image into its constituent regions or objects, so that the pixels in said regions have similar properties or attributes, *e.g.*, gray level, contrast, or texture.

The segmentation process is entrusted with evaluating each pixel in the image and deciding whether it contains characteristics of interest.

The Hough transform allows detecting curves in an image and is based on the search for geometric characteristics such as lines, triangles, and circular objects, among others. The Hough transform is one of the most widely used model-based segmentation techniques due to its robustness against noise and its behavior in the presence of holes in the object's border. When applying the Hough transform to an image, it is first necessary to obtain a binary image of the pixels that are part of the object boundary using threshold-based segmentation. The goal of this technique is to find points in the image that are aligned (Lestriandoko and Sadikin, 2016). Fig. 4 shows the use of this transform to find circles in an image during the final step of the detection process.

To process the images of the tiles placed on the board, a similar algorithm is employed, with the difference that there was no change of perspective regarding the image to be processed. The partial results of board processing are shown in Fig. 5.

Once all the tiles on the board have been identified, the system identifies the ones at the end and their number. To this effect, four axes are assigned (Fig. 6).

Then, the axes of each tile are compared against each other. When two axes have the same location within the image, it means that two tiles are joined. Therefore, if a tile has only a single axis shared with another tile, it means that it is a tile at the end of the current game. A combination of the above-mentioned axes is made to determine the tile side that indicates which end has to be played. For example, if the tile is vertical, the axes Y2 and X1 and X2 and Y1 are combined. If one of these combinations contains the shared axis, it is discarded. Therefore, the opposite side is chosen, whose number will be obtained, (Fig. 7).

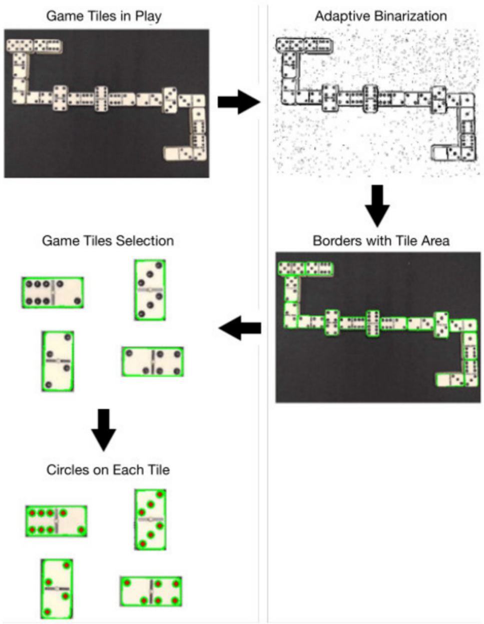

Figure 5. Algorithm for the detection of the game tiles placed on the board
Source: Authors

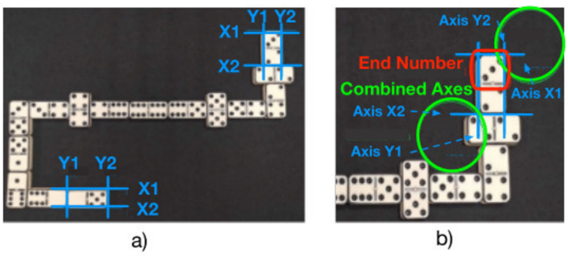

Figure 6. a) Axis assignment of both ends for the game tiles in play, b) end number example
Source: Authors

Decision system

A decision tree takes an object or situation as input (which is described through a set of attributes) and returns a decision, *i.e.*, the expected value of the output. The input attributes can be discrete or continuous. Thus, the decision tree develops a sequence of conditionals to reach a decision. Each internal node of the tree corresponds to a conditional for the value of one of the properties, and the branches that come out of the node are labeled with the possible values of said property. Each leaf node of the tree represents the value that must be returned upon reaching it.

For the intelligent system, the decision tree shown in Fig. 7 was used. In this tree, each node contains the main techniques or strategies according to their importance or priority in the game. These strategies have been compiled through research and collaboration from experienced dominoes players. Each of them is listed below, in order of priority.

Tile counting: The most important strategy is counting how many tiles of the same number are in hand and how many have already been placed on the board.

Blocking: Once the tile counting strategy has been executed, if the player has five or more tiles of the same number (adding both tiles in hand and on the board), the game should be played so that both ends match that number, as shown in Fig. 8.

Keeping ahead: After counting all the tiles in hand and those already placed on the board, if the player has six or seven tiles of the same number, the game should be played only at the other end, always keeping the tiles with that number until the end (if possible), as shown in Fig. 9.

One of each: When the player only has one option for playing on either side, they should play by placing the tile whose number has come out the most during the game.

Doubles first: Once the player has counted the tiles and determined that it is not possible to make a blocking move, they should get rid of the doubles first.

Best tile: If the player does not have doubles to place, the next strategy is called *the best tile*. To choose such a tile, the player first selects the number that has been placed the most on the board. Then, since a tile has two numbers, the player must choose the one that complements the already selected number with the one that exhibits the next majority of the tiles played, as shown in Fig. 10.

The first nodes of the decision tree correspond to the initial and basic plays of the game. A basic play is when a player only has one tile with a number that matches one of the ends of the tiles on the board. There are also situations when a player has two or more tiles that can be played. In that case, the doubles first or best tile rules apply. Once the tile to be played has been selected using this decision tree, the information is passed on for the execution system to physically perform the corresponding moves.

Execution system

To take the tiles and place them on the board, the robot uses its left arm. The joints involved in the movement are shown in Fig. 11. Here, the forward kinematic problem is used to find a transformation matrix that relates the coordinate system linked to the body in motion to a reference coordinate system. Homogeneous transformation matrices are used to achieve this representation, as shown in Eqs. (1) and (2), which includes translation and orientation operations. To this effect, a 4 x 4 matrix transforms a vector expressed in homogeneous coordinates from one coordinate system to another (Kofinas, Orfanoudakis, and Lagoudakis, 2013).

$$T = \begin{bmatrix} \text{rotation matrix} & \text{position vector} \\ f_{ix3} & \text{scaled} \end{bmatrix} = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$T = \begin{bmatrix} n & s & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

The vectors n, s, and a are unit orthogonal vectors, and p is a vector that describes the position x, y, and z of the origin of the current system with respect to the reference (Fig. 12).

Within the execution system, a forward kinematics model was created for the NAO robot's arm. With this model, the robot can take the tile selected for each move.

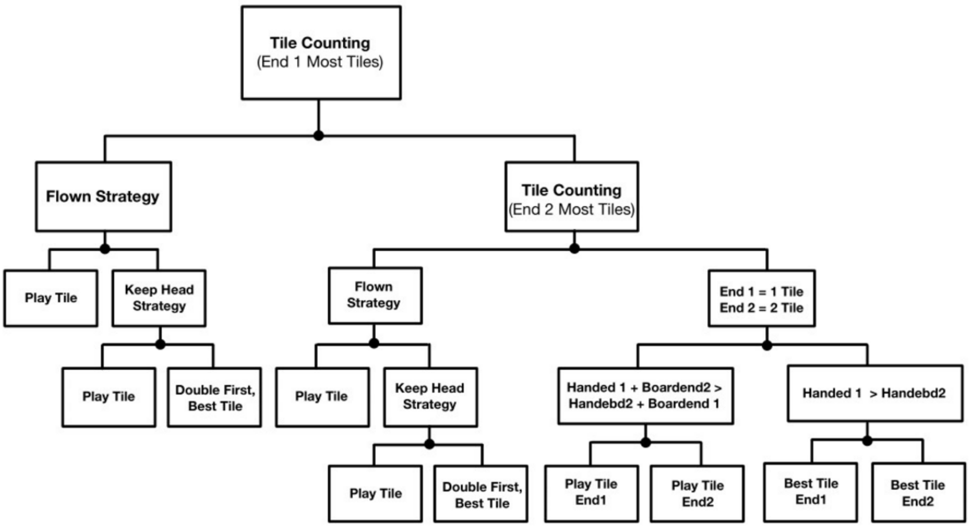

Figure 7. Decision tree for the intelligent game system
Source: Authors

Figure 8. Blocking play
Source: Authors

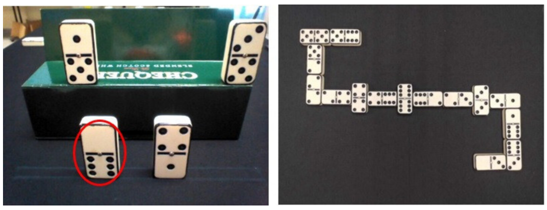

Figure 9. Keep-ahead play
Source: Authors

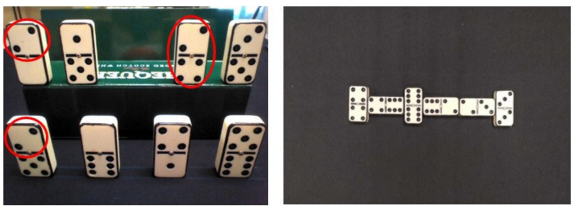

Figure 10. Best tile play
Source: Authors

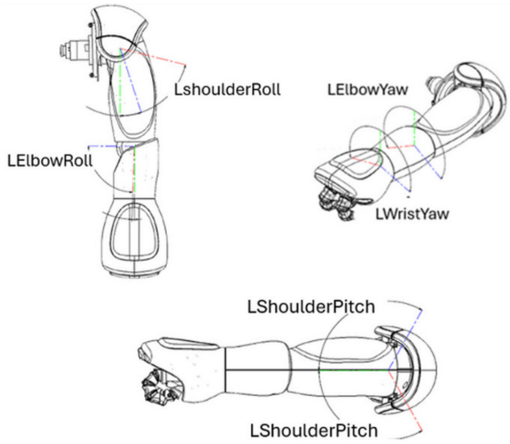

Figure 11. Joints used for the NAO robot's movement
Source: Authors

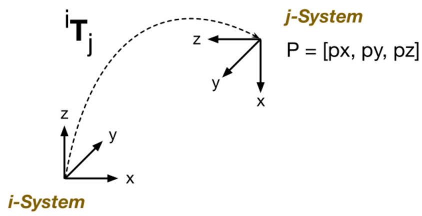

Figure 12. Reference coordinate system
Source: Authors

Axes must be assigned for each joint of the arm in order to establish this model, for which they must first be identified. Then, all the links that make up the robot's arm are listed, and the z-axis is assigned depending on the rotation of each joint. Afterwards, the x- and y-axes are located while following the right-hand rule. Once the axes have been properly assigned, the Denavit-Hartenberg (DH) parameters are obtained. These are the main basis for developing the forward kinematics model. The identification (numbering) or allocation of axle systems is first performed for each joint θ_i to obtain said parameters.

After assigning the axes, the DH parameters are determined as follows:

- θ_i: Angle around the Z_{i-1} axis, from the X_{i-1} axis to the X_i axis.
- d_i: Distance along the Z_{i-1} axis, from the origin of the $i-1$ system to the X_i axis.
- a_i: Distance along the X_i axis, from the Z_{i-1} axis to the Z_i axis.
- α_i: Angle around the X_i axis, from the Z_{i-1} axis to the Z_i axis.

Table 1. DH parameters

Joints	Parameter			
	a	α	d	θ
LShoulderPitch	0	$-\pi/2$	0	θ_1
LShoulderRoll	0	$\pi/2$	0	$\theta_2 - \pi/2$
LElbowYaw	0	$-\pi/2$	d_3	θ_3
LElbowRoll	0	$\pi/2$	0	θ_4
LWristYaw	0	$-\pi/2$	d_5	θ_5

Source: Authors

Based on the information in Table 1, the transformation matrix for each joint is obtained using the general matrix presented in Eq. (3).

$$\begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & \alpha_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & \cos\theta_i \sin\alpha_i & \alpha_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Then, Eq. (4) is applied to obtain the general transformation matrix of the entire kinematics model.

$$T^{1st\,link}_{end\,of\,vector} = T^0_1 + T^1_2 + T^2_3 + T^3_4 + T^4_5 + T^5_6. \quad (4)$$

The resulting matrix, shown in Eq. (5), contains the rotation and translation components of the NAO robot's arm with respect to the lower part of its torso (Fig. 13). The system outputs correspond to Px, Py, and Pz.

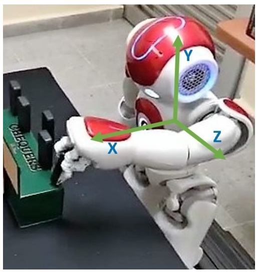

Figure 13. Reference coordinate system on the NAO robot's arm
Source: Authors

$$T^0_n = \begin{bmatrix} n_x & 0_x & a_x & P_x \\ n_y & 0_y & a_y & P_y \\ n_z & 0_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

For this specific case, the equations for each coordinate axis are as follows.

For the x-coordinate:

$$\begin{gathered} P_x = -d_3 \cos(\theta_1)\cos(\theta_2) \\ + d_5 \left[(-\sin(\theta_1)\sin(\theta_3) + \sin(\theta_2)\cos(\theta_1)\cos(\theta_3))\sin(\theta_4) \right. \\ \left. -\cos(\theta_1)\cos(\theta_2)\cos(\theta_4)\right]. \end{gathered} \quad (6)$$

For the y-coordinate:

$$\begin{gathered} P_y = -d_3 \sin(\theta_1)\cos(\theta_2) \\ + d_5 \left[(\sin(\theta_1)\sin(\theta_2)\cos(\theta_3) + \sin(\theta_3)\cos(\theta_1)\sin(\theta_4)) \right. \\ \left. -\sin(\theta_1)\cos(\theta_2)\cos(\theta_4)\right]. \end{gathered} \quad (7)$$

For the z-coordinate:

$$\begin{gathered} P_z = d_3 \sin(\theta_2) + d_5 \left(\sin(\theta_2)\cos(\theta_4) \right. \\ \left. + \sin(\theta_4)\cos(\theta_2)\cos(\theta_3)\right). \end{gathered} \quad (8)$$

Complete system and user interface

Once all the systems had been developed, they were put together in a single program using the Naoqi (Python) framework. Naoqi was selected because it provides all the libraries that allow access to the elements of the robot and its language recognition and processing functions. The latter are very important, as voice commands indicate the start and the end of the NAO robot's turn. The complete system takes less than 10 seconds to decide on and perform a play.

An interface to visualize the joint-system process of each play was devised using the Python Tkinter tool and an external computer. Through it, the visualization, processing, and decision system outputs could be observed (Fig. 14). The execution of the play, *i.e.*, the movements made to reach a tile, were visualized using the Coreographe software, which is part of the NAO robot's tools.

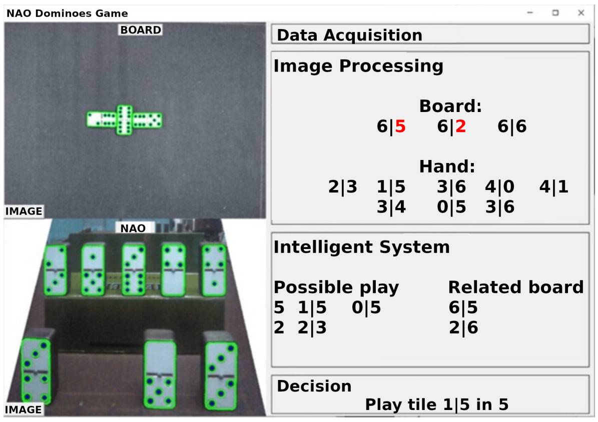

Figure 14. Complete system interface
Source: Authors

The interface was designed with two modes of visualization: (i) the process can be observed step by step, and (ii) the operation of each subsystem can be continuously visualized in an actual dominoes game.

Results and discussion

The testing setup comprised nine games of dominoes in a three-player variant (two humans, one robot), where the NAO robot interacted in a certain number of moves. If the move performed by the robot was incorrect, the conditions of the game before it were restored, and the interaction was repeated. For testing, the NAO robot was meant to interact in the most challenging situations, which is why the human players knew all the tiles in hand.

This section presents the data collected during the nine games played, classifying the errors reported by the vision, decision, and execution systems and counting the number of successful and unsuccessful plays for the complete system. A confidence interval of 90% was used in all cases.

Vision system

Table 2 lists the number of times that a play was incorrect due to bad image processing regarding the game tiles on the board (B) and in hand (H), *i.e.*, when the tile was incorrectly detected, resulting in an invalid move. Note that failures in the vision system were attributed to the image processing algorithms used. This table also shows the calculated percent error for each game and a total accumulated error after the nine games.

The number of times that there was an error when viewing the tiles on the board and in hand (X) was 5, and the total number of moves (n) was 89. Therefore, the value of the proportion (p) was 0,0562. Knowing the value of $Z_{0.95}$, an interval of 0.0562 ± 0.0402 was obtained.

Table 2. Results obtained for the vision, execution, and decision systems

Game	Moves	Error Vision			Execution		Decision	
		B	H	%	#	%	#	%
1	10	0	0	0	1	10	0	0
2	10	0	0	0	1	10	0	0
3	11	1	0	9.09	0	0	1	9.09
4	9	0	1	11.11	0	0	1	11.11
5	9	0	0	0	0	0	1	11.11
6	11	1	0	9.09	1	9.09	0	0
7	10	1	0	10	2	20	0	0
8	10	1	0	10	1	10	0	0
9	9	0	0	0	1	11.11	0	0
Total	89	4	1	5.62	7	7.87	3	3.37

Source: Authors

According to the results obtained, it can be said that the probability that the robot's vision system makes a mistake during a play is between 1.6 and 9.6%, with 90% confidence. Depending on the development of a full three-players game, the robot makes between 9 and 11 moves, which indicates that the robot can make vision-related mistakes at most once (if any). This error may be due to external factors, such as variations in the external lighting of the game scenario.

Decision system

For this system, the errors listed not only correspond to erroneous moves, but also to failures in selecting the most suitable tile at a given moment in the game. Table 2 shows the calculated percent error for each game and a cumulative total after the nine games.

The number of times that there was an error related to the intelligent system's decision on the tile to be played (X) was 3, and n had the same value: 89. Therefore, the value of the proportion (p) was 0.0337. Knowing the value of $Z_{0.95}$, an interval of 0.0337 ± 0.0315 was obtained.

According to the results, it can be said that the probability that the robot makes a decision-related mistake during a play is between 0.2 and 6.5%, with 90% confidence.

This subsystem's error probability is very low, which indicates that errors only occur in very specific cases. As previously described, since tile selection is entrusted to a decision tree that contains hierarchically organized game strategies, the robot may make an incorrect decision if the optimal play does not follow the hierarchy established with the help of highly experienced players.

Execution system

For this system, the errors listed correspond to the times that the selected tile was not taken or was incorrectly placed on the board. Table 2 shows the calculated percent error for each game and a cumulative total after the nine games.

The number of times that the robot made a mistake while taking the selected tile (X) was 7, and n was 89. Hence, the value of the proportion (p) was 0.0787. Knowing the value of $Z_{0.95}$, an interval of 0.0787 ± 0.0469 was obtained.

According to the results, it can be stated that the probability that the robot will not take the selected tile due to its kinematics model is between 3.2 and 12.6%, with 90% confidence.

Complete system

For the complete system, all the plays, both successful and unsuccessful, were counted, and the total performance of the system was calculated. Table 3 shows the results obtained by calculating the robot's efficiency in each game, as well as a cumulative total after the nine games.

Table 3. Results obtained for the complete system

Game	Moves	Move Successful	Unsuccessful	% Efficiency
1	10	9	1	90
2	10	9	1	90
3	11	9	2	81.82
4	9	7	2	77.78
5	9	8	1	88.89
6	11	9	2	81.82
7	10	7	3	70
8	10	8	2	80
9	9	8	1	88.89
Total	89	74	15	83.15

Source: Authors

For the complete system, the number of times that the robot made successful plays when interacting with the board game (X) was 74, and the total number of moves (n) was 89. Therefore, the value of the proportion (p) was 0.8315. As with the subsystems, knowing the value of $Z_{0.95}$, an interval of 0.8315 ± 0.0653 was obtained.

According to the results, it can be said that the probability that the robot makes a successful move during a game of dominoes is between 76.6 and 89.7%, with 90% confidence.

The correct operation of the complete system is contingent upon an absence of errors in its subsystems. Therefore, correctly processing the captured images, deciding on the optimal tile to be played, and properly taking and delivering the tile are a necessity. In this vein, the probability that the robot makes a successful move each time it interacts with the game is approximately eight to nine times out of 10.

The findings of this study are consistent with previous research on HRI in games. The proposed approach aligns other studies that have employed user-centered robot designs to improve the performance of HRI systems (Asfour *et al.*, 2018). For example, the results of refining the vision system (Zhu *et al.*, 2018; Juang and Zhang, 2019; Yan *et al.*, 2022), improving the robot's decision-making (Díaz *et al.*, 2018; Goenaga *et al.*, 2020), and experimenting with different players and humanoid robots (Wei, 2020; Zhai *et al.*, 2017). Overall, this study contributes to the growing body of research on HRI entertainment (specifically, board games) and highlights the potential for humanoid robots to interact with humans in various settings (Lin *et al.*, 2022; Inoue *et al.*, 2022). According to the experimental results, further research is needed to refine the system and improve its performance in real-world settings while considering the complexity of the game, the robot's level of autonomy, and the quality of the algorithms used.

Conclusions

This paper provides an overview of research on the use of humanoid robots in a variety of fields, including education, therapy, and entertainment. One of the most important contributions of this study is the description of an intelligent system for playing dominoes with a NAO robot. The system includes image processing algorithms for detecting and identifying tiles on the board and in hand, a decision tree algorithm for determining the best move, and a forward kinematics model for physically moving the robot's arm to perform the selected move. The system also includes a user interface for process visualization and monitoring.

The proposed system successfully enables a humanoid robot to interact with people by executing plays in a game of dominoes. The case studies considered include actual game situations, in which the robot correctly interacts with other players based on conditions provided as the game progresses. Thanks to the proposed approach, the robot can select the most appropriate tile, which entails a friendlier HRI. The proper operation of the entire system is dependent on the absence of errors in its subsystems. Thus, correctly processing the captured images, determining the most tile to be played, and properly taking and placing tiles on the board are required.

According to the findings and the analysis presented in this paper, the vision system has an average error of 5.62%, the decision system reports 3.37%, and the value for the execution system is 7.87%. When all the components function properly, the NAO robot can complete successful plays at an 83% rate. Each HRI-enabled system was validated and comprised various subsystems that executed particular functions based on the application's requirements, with each subsystem's error being minimized to enhance overall system performance.

The vision system of a robot can be affected by various factors such as changes in external lighting conditions, player hand occlusions, or quick movements, which can result in image processing errors. Additionally, hardware or software limitations can further impair the robot's ability to accurately interpret and respond to visual cues, which can increase the likelihood of errors.

The decision system helps to make optimal plays with its structured decision tree framework, but deviations from predefined strategies are sometimes necessary due to unforeseen game states or unexpected player actions. Relying solely on historical data or predefined rules may overlook nuanced situational factors, leading to suboptimal decisions. To maximize success, strategies must be adapted accordingly.

The execution system is vulnerable to mechanical issues, such as imprecise motor control or workspace obstructions, that could compromise its reliability. Furthermore, communication delays between subsystems may hinder the timely execution of planned actions, leading to suboptimal gameplay performance. It is crucial to consider these factors for an optimal system performance.

After analyzing various factors, it is clear that the NAO robot's ability to play dominoes is dependent on addressing a number of technical challenges and environmental variables. By thoroughly assessing these factors and their impact on subsystem functionality, we can develop specific strategies to improve the robot's overall reliability and reduce errors during gameplay.

Finally, this study demonstrates the potential of humanoid robots in promoting and improving HRI in various settings. The development of intelligent systems that can interact with people and plan and execute their moves in board games represents a significant advancement in this field of study. The proposed method demonstrates that the robot can successfully play a board game with people by combining image processing, decision-making, and physical movement systems, making the HRI more effective. Further research could improve the system's performance by refining the vision system, experimenting with different players and humanoid robots, and exploring other intelligent algorithms.

Author contributions

Conceptualization: *Alex Medina* and *Christian G. Quintero M.*. Data curation: *Alex Medina*, *Daniela Charris*, and *Christian G. Quintero M.*. Formal analysis: *Alex Medina*, *Daniela Charris*, *Mauricio Pardo*, and *Christian G. Quintero M.*. Funding acquisition: *Alex Medina* and *Christian G. Quintero M.*. Investigation: *Alex Medina*, *Daniela Charris*, *Mauricio Pardo*, and *Christian G. Quintero M.*. Methodology: *Mauricio Pardo* and *Christian G. Quintero M.*. Project administration: *Christian G. Quintero M.*. Resources: *Mauricio Pardo*. Supervision: *Mauricio Pardo* and *Christian G. Quintero M.*. Validation: *Daniela Charris* and *Mauricio Pardo*. Visualization: *Mauricio Pardo*. Writing - original draft: *Alex Medina*, *Daniela Charris*, *Mauricio Pardo*, and *Christian G. Quintero M.*. Writing - review and editing: *Alex Medina*, *Daniela Charris*, *Mauricio Pardo*, and *Christian G. Quintero M.*.

References

Ajani, H., Obasekore, B., Kang, and Rammohan, M. (2023). Robotic assistance in radiology: A covid-19 scenario. *IEEE Access*, *11*, 49785–49793.

Asfour, et al. (2018). Armar-6: A collaborative humanoid robot for industrial environments. In *2018 ieee-ras 18th int. conf. humanoid robots* (*humanoids*) (pp. 447–454).

Barakeh, S., Alkork, A., Karar, S., Said, S., and Beyrouthy, T. (2019). Pepper humanoid robot as a service robot: A customer approach. In *2019 3rd int. conf. bio-eng. smart tech.* (*biosmart*) (pp. 1–4).

Barnes, J., FakhrHosseini, S., Vasey, E., Ryan, J., Park, C., and Jeon, M. (2019). Promoting steam education with child-robot musical theater. In *2019 14th acm/ieee int. conf. human-robot interact.* (*hri*) (p. 366).

Bi, Q., Yang, M., Zhang, A., Goupil, A., and Feng, L. (2018). Accurate football detection and localization for nao robot with the improved hog-svm approach. In *2018 chinese autom. cong.* (*cac*) (pp. 567–571).

Budiharto, W., Cahyani, A. D., Rumondor, P. C. B., and Suhartono, D. (2017). Edurobot: intelligent humanoid robot with natural interaction for education and entertainment. *Procedia Comp. Sci.*, *116*, 564–570.

Cao, J., et al. (2020). Robot-assisted joint attention: A comparative study between children with autism spectrum disorder and typically developing children in interaction with nao. *IEEE Access*, *8*, 1–1.

Díaz, J., Shaik, J., Santofimio, J., and Quintero, M. (2018). Intelligent execution of behaviors in a nao robot exposed to audiovisual stimulus. In *2018 ieee 2nd colombian conf. robotics automat.* (*ccra*) (pp. 1–6).

Feidakis, I., Gkolompia, A., Marnelaki, K., Marathaki, S., Emmanouilidou, S., and Agrianiti, E. (2023). Nao robot, an educational assistant in training, educational and therapeutic sessions. In *2023 ieee global eng. educ. conf.* (*educon*) (pp. 1–6).

Goenaga, L., Navarro, C., Quintero, M., and Pardo, M. (2020). Imitating human emotions with a nao robot as interviewer playing the role of vocational tutor. *Electronics*, *9*.

Hu, F., Zhao, J., Meng, and Wu, S. (2020). Application of deep reinforcement learning in the board game. In *2020 ieee int. conf. info. tech. big data art. intel.* (*iciba*) (Vol. 1, pp. 809–812).

Inoue, F., Jimenez, T., Haruta, M., and Oonuki, M. (2022). Effect of impression on learners during partnered robots learning programming while playing board games against each other. In *2022 joint 12th int. conf. soft comp. intel. syst. and 23rd int. symp. adv. intel. syst.* (*scis&isis*) (pp. 1–4).

Jeon, M., et al. (2017). Robot opera: A modularized afterschool program for steam education at local elementary school. In *2017 14th int. conf. ubiquitous robots ambient intel.* (*urai*) (pp. 935–936).

Juang. (2022, Jan.). Humanoid robots play chess using visual control. *Multimedia Tools App.*, *81*, 1–22.

Juang, and Zhang, J. (2019). Visual tracking control of humanoid robot. *IEEE Access*, *7*, 29213–29222.

Karmanova, V., Serpiva, S., Perminov, A., Fedoseev, A., and Tsetserukou, D. (2021). Swarmplay: Interactive tic-tac-toe board game with swarm of nano-uavs driven by reinforcement learning. In *2021 30th ieee int. conf. robot human interac. comm.* (*ro-man*) (pp. 1269–1274).

Karunanayake, et al. (2020). Towards a smart opponent for board games: Learning beyond simulations. In *2020 ieee int. conf. syst. man cyber.* (*smc*) (pp. 1943–1950).

Knox, and Watanabe, K. (2018). Aibo robot mortuary rites in the japanese cultural context. In *2018 ieee/rsj int. conf. intel. robots syst.* (*iros*) (pp. 2020–2025).

Kofinas, E., Orfanoudakis, M., and Lagoudakis, M. (2013). Complete analytical inverse kinematics for nao. In *2013 13th int. conf. autonom. robot syst.* (pp. 1–6).

Kołosowski, A., Wolniakowski, K., and Miatliuk, K. (2020). Collaborative robot system for playing chess. In *2020 int. conf. mechatronic sys. mat.* (*msm*) (pp. 1–6).

Lestriandoko, and Sadikin, R. (2016). Circle detection based on hough transform and mexican hat filter. In *2016 int. conf. control infor. app.* (*ic3ina*) (pp. 153–157).

Li, E., Imeokparia, M., Ketzner, M., and Tsahai, T. (2019). Teaching the nao robot to play a human-robot interactive game. In *2019 int. conf. comp. sci. comp. intel.* (*csci*) (pp. 712–715).

Lin, S., Ng, and Sebo, S. (2022). Benefits of an interactive robot character in immersive puzzle games. In *2022 31st ieee int. conf. robot human interac. comm.* (*ro-man*) (pp. 37–44).

Lobos-Tsunekawa, F., Leiva, S., and Solar, J. (2018). Visual navigation for biped humanoid robots using deep reinforcement learning. *IEEE Robotics Automat. Lett.*, *3*, 3247–3254.

Mercier, and Lubart, T. (2021, Feb.). The effects of board games on creative potential. *J. Creat. Behavior*, *55*.

Mohan, and Kuchenbecker, K. (2019). A design tool for therapeutic social-physical human-robot interactions. In *2019 14th acm/ieee int. conf. human-robot interac.* (*hri*) (pp. 727–729).

Moya, E., Slawiñski, V., Mut, B., and Wagner, B. (2021, May). Intercontinental bilateral-by-phases teleoperation of a humanoid robot. *IEEE Latin Amer. Trans.*, *20*, 64–72.

Ovalle-Magallanes, R., et al. (2021). Transfer learning for humanoid robot appearance-based localization in a visual map. *IEEE Access*, *9*, 6868–6877.

Patil, D., Fegade, A., Kadam, P., Patil, N., and Singhaniya, N. (2021). A novel framework for robotic chess. In *2021 2nd int. smart elec. comm.* (*icosec*) (pp. 1–6).

Piperakis, M., Koskinopoulou, A., and Trahanias, P. (2018). Nonlinear state estimation for humanoid robot walking. *IEEE Robotics Automat. Lett.*, *3*, 3347–3354.

Raghavan, H. M. B., Srinivasan, R., Dey, S., and Chandar, T. (2021). Automated laser alignment and image processing-based robotic carrom player. In *2021 20th int. conf. adv. robotics* (*icar*) (pp. 499–504).

Rath, N., Mahapatro, P., Nath, P., and Dash, R. (2019). Autonomous chess playing robot. In *2019 28th ieee int. conf. robot human interac. comm.* (*ro-man*) (pp. 1–6).

Schadenberg. (2019). Predictability in human-robot interactions for autistic children. In *2019 14th acm/ieee int. conf. human-robot interac.* (*hri*) (pp. 748–750).

Sun, C., Wang, T., Zheng, H., and Liu, H. (2023). An improved object detection method based on nao robot. In *2023 ieee 3rd int. conf. power elec. comp. app.* (*icpeca*) (pp. 1184–1188).

Tres. (2014). *Estudio e implementación de algoritmos de resolución del juego del dominó para un robot antropomórfico* (Unpublished master's thesis). Universidad Politécnica de Cataluña, Spain.

Wang, X., Xue, B., and Chen, B. (2020). Matsuoka's cpg with desired rhythmic signals for adaptive walking of humanoid robots. *IEEE Trans. Cyber.*, *50*, 613–626.

Wei. (2020). A comprehensive approach to the generation of human-like arm movements on robot nao. *IEEE Access*, *8*, 172869–172881.

Yan, S., Li, C., Liu, M., Liu, M., and Chen, Q. (2022). Roboseg: Real-time semantic segmentation on computationally constrained robots. *IEEE Trans. Syst. Man Cyber. Syst.*, *52*, 1567–1577.

Yang, M., Shyu, H., Yu, S., Sun, N., Yin, W., and Chen, W. (2019). Integrating image and textual information in human–robot interactions for children with autism spectrum disorder. *IEEE Trans. Multimedia*, *21*, 746–759.

Zhai, S., Wen, J., Zhu, J., and Guo, G. (2017). Trajectory planning of nao robot arm based on target recognition. In *2017 int. conf. adv. mechatronic syst.* (*icamechs*) (pp. 403–407).

Zhu, H., Yi, R., Chellali, and Feng, L. (2018). Object recognition and localization algorithm based on nao robot. In *2018 27th ieee int. symp. robot human interac. comm.* (*ro-man*) (pp. 483–486).