

A Voice-Controlled Chess Game That Helps Physically Handicapped People Play Chess without Using Their Hands

N Likitha¹, Lakshaditi J², Kusuma N.S³, Kumkum Rathore⁴, Nagendra R⁵

¹²³⁴Student, USN:1MV23CS095, Dept of Computer Science and Engineering, Sir M Visvesvaraya Institute of Technology, Bengaluru, Karnataka, India

⁵Guide, Dept of Computer Science and Engineering, Sir M Visvesvaraya Institute of Technology, Bengaluru, Karnataka, India

Abstract - Digital chess applications commonly depend on a mouse, keyboard, or touchscreen to select pieces and squares. These interaction methods can create barriers for people with limited hand mobility, reduced fine-motor control, or other physical conditions that make repeated pointer actions difficult. This paper describes a voice-controlled chess application intended to provide an alternative, hands-free method of entering moves. The system combines speech-to-text conversion, chess-command interpretation, rule-based move validation, game-state management, an optional computer opponent, and spoken feedback. A voice command is treated as a request rather than an instruction that immediately changes the board: the recognized text is parsed into a candidate move, checked against the current position, and executed only when it is legal. Ambiguous, unrecognized, or illegal commands leave the board unchanged and prompt the user to try again. The application retains a visual chessboard and optional conventional input so that users can choose an interaction method that suits their needs. The design also supports play against a chess engine, identified in the project documentation as Stockfish, with the response move announced through audio feedback. The manuscript describes the architecture, processing workflow, implementation responsibilities, evaluation measures, and current limitations. Quantitative accuracy and response-time results are not reported because completed test measurements were not supplied. The paper therefore presents a prototype design and a transparent evaluation plan rather than unsupported performance claims.

Key Words: Voice-first-interface, Multimodal interaction, Accessible interface, Hand-free chess, Interactive chess system, Human-AI interaction, Human-computer interaction, Real-time interaction

1. INTRODUCTION

Chess is a strategy game in which players repeatedly select pieces, identify destination squares, and execute moves while following a defined set of rules. In a physical game, these actions are performed by moving pieces on a board. In a digital game, the same interaction is commonly carried out with a mouse, keyboard, or touchscreen. Although these controls are familiar and efficient for many players, they can be difficult to use for people who have limited hand mobility, involuntary movement, reduced dexterity, or difficulty sustaining repeated fine-motor actions. An interface that offers only pointer-based movement can therefore become a barrier to participation, even when the strategic part of the game is accessible.

Speech provides a possible alternative input channel. Instead of selecting a piece and then selecting a square, a player can describe the intended move aloud. However, adding a microphone to a chess application is not sufficient on its own. Spoken language is variable: a command may be misheard, may omit important information, or may describe a move that is illegal in the current position. The application must distinguish between recognizing words and deciding whether those words describe an executable chess move.

This work presents a voice-controlled chess prototype designed around that distinction. The user provides a spoken move, the speech-recognition component produces a transcript, and the command parser converts the transcript into a structured request. The chess logic then checks the request against the current game state. Only a legal and sufficiently unambiguous move is applied. When the command cannot be understood or is not legal, the application keeps the current position and gives the player audio feedback. This conservative approach is important because a mistaken move can affect every later decision in a chess game.

The application also supports a computer-opponent mode. After a player's move is accepted, the current position is supplied to the chess engine, which chooses a reply according to the configured playing level. The interface then updates the board and announces relevant game events. A conventional visual board and mouse input are retained as optional interaction paths rather than being removed entirely. This makes the application adaptable to users who prefer a mixture of voice and visual controls.

The principal contribution is the integration of voice input, command interpretation, chess-rule validation, game-state tracking, computer-opponent play, and spoken feedback in one application workflow. The paper focuses on the system design and the responsibilities of its components. It does not claim that the prototype has been validated through a large-scale accessibility study, and it does not report performance values for which test data are unavailable.

The rest of the paper is organized as follows. Section II discusses related work represented in the original project references. Section III defines the problem and objectives. Section IV describes the proposed architecture and workflow. Section V explains the implementation stages and move-processing logic. Section VI outlines an evaluation methodology and discusses the prototype's observable behavior and limitations. Section VII considers accessibility, security, and practical deployment concerns. Section VIII identifies future work, and Section IX concludes the paper.

1.1 RELATED WORK

Speech-based interaction has been explored as an alternative to conventional controls in games and interactive applications. The work cited in the project manuscript covers voice-controlled gaming, speech recognition, and voice-based chess. These areas are relevant because a speech-controlled chess interface must combine general speech processing with the specific state and rules of a chess game.

Kolipaka's work on voice-based chess discusses the use of voice recognition as a way to issue chess commands and identifies practical concerns such as command recognition, differences in accents and speech patterns, and responsiveness [1]. These concerns apply directly to a system in which the destination square and piece type must be interpreted accurately. The present design addresses the distinction between a recognized phrase and a legal chess action by placing a validation stage between command interpretation and board updates.

Strzałko examines approaches and challenges associated with speech recognition and voice control in digital games [2]. The paper is relevant to the broader interaction design because a game must coordinate speech-to-text conversion with the logic that determines what an input means. A general recognizer may produce a plausible transcript without knowing whether that transcript is a valid command in a particular game. Consequently, speech recognition and game logic should remain separate modules with a clear interface between them.

Varma et al. discuss voice-controlled gaming tools in the context of learning and inclusive interaction [3]. Their topic supports the motivation for considering speech as an additional way to interact with software. A separate reference in the original manuscript also describes voice-controlled gaming tools [4]. The bibliographic details of that entry are retained from the supplied paper; the overlap in its title with [3] should be checked against the original publications before submission to ensure that the citation is accurate and not a duplicate or incorrectly attributed source.

Taken together, the cited studies motivate voice input as a useful interaction option but also indicate that recognition errors and command ambiguity must be addressed. The current application builds on that general direction by organizing voice control as a multi-stage process: transcription, command parsing, legality checking, state update, and feedback. Its intended contribution is this integrated workflow in the context of chess, rather than a claim that speech recognition or voice-controlled gaming is itself new.

1.2 PROBLEM DEFINITION AND OBJECTIVE

A. Problem Statement

The problem addressed by this project is the dependence of common digital chess interfaces on physical pointing or touch actions. Users with restricted hand movement may find it difficult to select a piece and its destination square repeatedly throughout a game. The system should provide a voice-based alternative without compromising the integrity of the chess position when a command is misrecognized or cannot be interpreted.

B. Design Objectives

The prototype is organized around the following objectives:

1. Accept spoken chess-move requests as an alternative to pointer-based input.
2. Convert speech into text and interpret the transcript as a structured move request.

3. Validate candidate moves against the current board position before execution.
4. Preserve the game state when a command is unclear or illegal.
5. Support a computer opponent through a chess engine when that mode is selected.
6. Communicate accepted moves, errors, and important game events through audio feedback.
7. Retain the visual board and optional conventional input to support different interaction preferences.

C. Scope and Contribution

The paper describes a prototype and its software workflow. The available project manuscript identifies Stockfish as the chess engine and describes speech-to-text processing and text-to-speech feedback, but it leaves the exact speech-recognition library or API unspecified. The implementation details that depend on that missing information are therefore described at the component level. The contribution is the system design and integration strategy; comparative usability results, measured recognition accuracy, and clinical accessibility claims are outside the evidence currently available.

2. PROPOSED SYSTEM AND METHODOLOGY

2.1 System Overview

The proposed system accepts input through voice or an optional mouse-based interface. Voice input is converted into a transcript, which is processed into a candidate move. The chess logic checks the move against the current position and updates the board only when the move is legal. In AI mode, the chess engine generates a response after the user's move is accepted. The interface presents the updated position and provides audio feedback about the action or any problem encountered.

This design separates four concerns that should not be conflated: capturing speech, interpreting a chess command, enforcing chess rules, and presenting the result. The separation allows an error at one stage to be handled without automatically corrupting later stages. For example, a transcript may contain a destination square but omit the piece type. The parser can identify the request as incomplete and ask for clarification instead of attempting to move a piece based on a guess.

2.2 System Architecture

The system architecture consists of the following logical layers:

1. Presentation Layer: Contains the chessboard, game settings, and voice or mouse input.
2. Processing Layer: Receives input and coordinates speech recognition, command interpretation, and the game controller.
3. Chess-Logic Layer: Maintains the current position, validates moves, and manages turn order.
4. AI Engine: When AI mode is enabled, the engine produces a response move based on the current position and configured difficulty.
5. Output Layer: Refreshes the board and communicates the result through audio.
6. Game Data: Maintains move history and game results.

The layers are logical responsibilities rather than a claim that each one runs as a separate process. Their separation is useful for maintenance and testing: the command parser can be tested with text transcripts, the chess logic can be tested with legal and illegal positions, and the feedback component can be checked independently for clarity. The architecture also allows the speech-recognition provider to be changed without requiring the chess rules to be rewritten, provided the recognizer continues to return a transcript in the expected form.

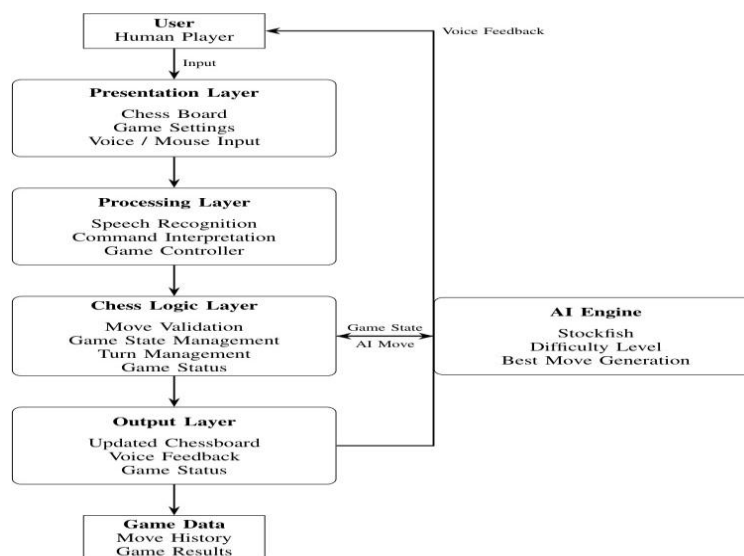


Fig. 1. System Architecture of the Proposed Voice-Based Chess System

2.3 Overall Workflow

The workflow begins when the user selects game settings and chooses voice or mouse input. A spoken command passes through speech recognition and command interpretation before it reaches chess-move validation. If the move is invalid or the command cannot be resolved, the system gives feedback and requests another command. If the move is valid, the game state is updated. When an AI opponent is enabled, the engine generates a reply and the state is updated again. The system then provides feedback and checks whether the game has ended.

The rejection path is as important as the successful path. A speech recognizer may return a transcript even when it has interpreted the utterance incorrectly. For this reason, the system should not modify the board at the point where text is produced. Instead, a candidate move must pass both interpretation and legality checks. This principle reduces the chance that recognition errors will directly alter the game

2.4 Data Flow

The data flow diagram presents the movement of information through the application. A voice command travels from the user to the speech-recognition component, which produces recognized text. The command-processing component converts that text into a move request, and the chess logic evaluates it against the current position. Game-state data and move history are updated after accepted actions. In AI mode, the engine receives the current position and returns a response move. Voice feedback communicates the result to the player.

The data flow highlights that the transcript is not the final representation of a move. The parser must produce a structured request, and the chess logic must determine whether that request can be applied to the current state. The state is the shared reference for the human player, the AI opponent, and the move-history component. Keeping one authoritative state avoids inconsistent board representations between the interface and the engine.

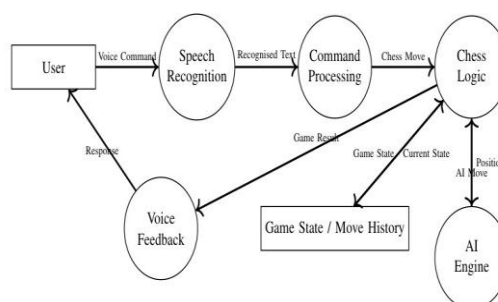


Fig. 3. Level-1 Data Flow Diagram of the Proposed Voice-Based Chess System

3. EVALUATION METHODOLOGY AND DISCUSSION

3.1 Evaluation Status

The proposed system is intended to support chess gameplay through voice commands. Its evaluation should examine speech recognition, command interpretation, move validation, response time, and audio feedback. However, the available manuscript does not provide measured accuracy, response-time values, or detailed experimental results. Therefore, numerical performance claims cannot be reported.

The system is designed to prevent invalid moves from being applied to the chessboard and to provide feedback when a command cannot be executed. These behaviours should be verified through systematic testing.

3.2 Evaluation Measures

The following measures can be used to evaluate the system:

1) Command Recognition Accuracy: Recognition accuracy measures the percentage of tested commands that are correctly recognized and interpreted.

$\text{Accuracy (\%)} = (\text{Number of Correctly Processed Commands} / \text{Total Number of Tested Commands}) \times 100$

2) Response Time: Response time measures the interval between the user's spoken command and the system's corresponding response. The measurement boundaries should be defined consistently during testing.

3) Invalid Command Handling: This measure checks whether illegal, unclear, or unsupported commands are rejected without changing the current chessboard state.

4) Audio Feedback: This measure evaluates whether the system provides understandable feedback when a move is accepted, rejected, or requires clarification.

Testing should include different command phrasings, speakers, microphone distances, and background-noise conditions.

3.3 Discussion of System Behaviour

The proposed workflow combines speech recognition, command interpretation, chess-rule validation, game-state management, and audio feedback. Separating these functions helps ensure that a recognized speech command is not executed automatically without checking its meaning and legality.

When a command is invalid or ambiguous, the system should retain the existing board state and provide appropriate feedback. This approach helps reduce unintended moves during gameplay.

However, the available information does not establish measured recognition accuracy, response time, or usability outcomes. These aspects require experimental validation before the system's overall performance can be quantified.

3.4 Limitations and Threats to Validity

The performance of a voice-controlled chess system may depend on microphone quality, background noise, speech clarity, and the speech-recognition configuration. Ambiguous chess notation may also lead to incorrect command interpretation.

The available manuscript does not provide a complete test dataset or quantitative evaluation results. Furthermore, the extent of testing for special chess rules, including castling, pawn promotion, and en passant, is not documented. These limitations should be considered when interpreting the system's capabilities.

4. ACCESSIBILITY AND PRACTICAL CONSIDERATIONS

4.1 Accessibility Benefits

The voice-controlled chess system aims to make digital chess more accessible to people with physical disabilities who may find repeated mouse or touchscreen interactions difficult. By allowing users to issue chess moves through spoken commands, the system can reduce dependence on conventional input methods.

4.2 User Interaction and Feedback

The system should provide clear feedback to indicate whether a spoken command has been recognized, interpreted, and accepted. Short audio confirmations can help users understand the current status of their moves. When a command is unclear or invalid, the system should communicate the issue and allow the user to try again.

Visual feedback may also be retained as a supplementary feature for users who can use a screen. Providing more than one feedback method can improve clarity during gameplay.

4.2 Practical Limitations

Voice interaction may not be suitable for every user or environment. Background noise, unclear pronunciation, microphone quality, and speech-recognition errors can affect command processing. Therefore, conventional input methods may be retained as an alternative where appropriate.

4.3 Privacy Considerations

If speech recognition uses an external service, audio recordings or recognized text may be processed outside the user's device. The implementation should clearly document how voice data is handled and obtain appropriate user consent. The available manuscript does not specify whether speech processing is performed locally or through a cloud service; therefore, the actual data-handling method requires clarification.

5. FUTURE WORK

The proposed voice-controlled chess system can be improved in several ways to enhance its accessibility, reliability, and overall user experience.

Future work can focus on improving speech recognition under different background-noise conditions and supporting different speaking styles. The command interpretation module can also be enhanced to handle ambiguous commands and provide clearer feedback when additional information is required.

Further testing can be conducted to evaluate command recognition accuracy, response time, invalid-move handling, and the overall usability of the system. The testing environment, software versions, and evaluation procedure should be documented to make the results reproducible.

Additional improvements may include testing special chess moves, such as castling, pawn promotion, and en passant, along with game-ending conditions such as checkmate and stalemate. These features should be verified through systematic testing.

Finally, usability studies involving people with different physical abilities can help identify practical challenges and improve the accessibility of the application. Support for additional languages and improved voice interaction may also be explored in future versions.

6. CONCLUSION

The proposed voice-controlled chess system aims to make digital chess more accessible to people with physical disabilities by enabling them to play using spoken commands instead of relying entirely on conventional input devices. The system combines speech recognition, command interpretation, chess-rule validation, game-state management, and audio feedback to support the gameplay process.

A key aspect of the proposed approach is that spoken commands must be interpreted and validated before a move is executed. Invalid or unclear commands should not change the current chessboard state, and appropriate feedback should be provided to the user.

The system offers a potential approach to improving accessibility in digital chess. However, further experimental evaluation is required to measure recognition accuracy, response time, reliability, and usability. Future improvements and user testing can help establish the system's effectiveness and make it more suitable for people with different accessibility needs.

REFERENCES

- [1] M. Kolipaka, "Voice-Based Chess: Enhancing User Experience through Voice Recognition Technology," *International Journal of Research Publication and Reviews*, vol. 5, no. 6, pp. 488–493, Jun. 2024.
- [2] D. Strzałko, "Voice Controlled Games – The Approach and Challenges of Implementing Speech Recognition and Voice Control in Games," in *Proceedings of the Federated Conference on Computer Science and Information Systems*, vol. 26, pp. 229–230, 2021, doi: 10.15439/2021F143.
- [3] D. B. Varma, D. Pragnya Sree, and M. Roopa Sree, "Voice-Controlled Gaming Tools for Enhanced Learning in the Skill Ecosystem," *International Journal of Research Publication and Reviews*, vol. 6, no. 4, pp. 14281–14285, Apr. 2025, doi: 10.55248/gengpi.6.0425.1660.
- [4] A. B., S. N., B. A., H. Govinda Raj, and A. Y., "Voice Controlled Gaming Tools for Enhanced Learning in the Skill Ecosystem," *IOSR Journal of Computer Engineering*, vol. 27, no. 3, ser. 1, pp. 82–92, May–Jun. 2025, doi: 10.9790/0661-2703018292.