

Article

Not peer-reviewed version

---

# Development of a Large Language Model-Based System For Enhanced Access to Chess for Visually Impaired Individuals

---

[Arko Chakrabartiroy](#)\*

Posted Date: 7 October 2025

doi: 10.20944/preprints202510.0447.v1

Keywords: chess; large language models; speech-to-text models; machine learning; chess AI



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

# Development of a Large Language Model-Based System For Enhanced Access to Chess for Visually Impaired Individuals

Arko Chakrabartiroy

Stuyvesant High School, New York, NY 10282; achakrabartiroy@gmail.com

## Abstract

Chess is one of the miraculous inventions of mankind. However, traditional chess interfaces often pose significant challenges for individuals with visual impairments, limiting their ability to engage with the game independently. The objective of this paper is to develop a computerized system where a visually impaired person can play chess with a machine, with another person, visually impaired or not, and train with chess puzzles. It leverages recent advances in the development of machine learning models, most notably the latest versions of large language models (LLMs), speech-to-text conversion models, and chess AI models, to build a cohesive system. We developed a system that initially converts audio input from the visually impaired user to text representation, leveraging an audio-to-text machine learning (ML) model and an LLM. Subsequently, the text representation is fed to chess AI, Stockfish, which generates the optimal countermove. Finally, the optimal counter-move from Stockfish is passed through a text-to-speech model that is played as audio to the user. The proof-of-concept system developed in this study can make chess accessible to visually impaired people by enabling them to engage with the game in a way that is natural and intuitive.

**Keywords:** chess; large language models; speech-to-text models; machine learning; chess AI

---

## 1. Introduction

Chess is a sport that has multifaceted benefits – it brings joy, refreshes the mind, builds community, and fosters analytical/intellectual growth (Yang, 2025). Moreover, chess builds resilience and confidence on the one hand, and concentration and discipline on the other (Chase and Simon, 1973). In short, chess can be life-changing and has long-lasting benefits (Ross, 2006). But the blessing of chess is out of reach for many people – and many children - in this world. Just focusing on the U.S., about 20 million Americans – 8 percent of the U.S. population – have visual impairments (Health Policy Institute, 2008). This includes an agonizingly large number of children – about 600,000 children under the age of 18 have vision difficulty (American Foundation for the Blind, 2023).

In recent years, there has been significant interest in harnessing the power of technology to improve the quality of life for people with disabilities. For a comprehensive review of these efforts for making board games accessible to people with visual impairments, see Agrimi et al. (2024). This interest has coincided with the rapid development of machine learning, and researchers have worked to design an interface aimed at helping visually impaired people play board games via improved interaction with a normal or touch screen (Gnecco et al, 2024), but to date there has not been research on leveraging the recent advances in artificial intelligence (AI) and machine learning (ML) specifically to make chess more accessible for the visually impaired. Separately, there have been some isolated instances in the last few years of leveraging technology to make it easier for visually impaired people to play chess. For example, a project at the University of North Carolina employs a software package to allow visually impaired people to learn and play chess fully without vision, using only a keyboard interface (Accessible Chess Tutor (ACT)). Similarly, the chess app embedded in Monarch, a multiline

braille device that incorporates tactile graphics integrated with braille, has the potential to promote tactile discrimination and encourage peer-to-peer relationships (American Printing House, 2024).

However, while the ACT project and the Monarch Chess App both show demand for a practical device that would satisfy both educational and recreational needs for users with visual disabilities, neither of these studies leveraged recent AI advances to help improve access for visually impaired individuals to chess. Moreover, such a specialized device may not be easily accessible to the visually impaired and may not be available in resource-limited settings. These systems also bring with them a high learning barrier, as Braille and other mechanical means of taking in user input are unintuitive. In contrast, a computerized audio system has the potential for being easily accessible anywhere in the world and allows users to communicate in a way they would be most comfortable, speech. While some studies have discussed the potential for using machine learning to benefit visually impaired people, they fall short of providing concrete solutions that would aid visually impaired people in playing chess in a natural way. Additionally, with recent advancements in generative AI transformer-based models, there is significant potential to leverage LLMs, along with other artificial intelligence and machine learning models, to enhance accessibility and user experience for visually impaired users of chess. The current paper seeks to use these recent developments to fill this important gap.

In this study, we use a Python platform along with generative AI and ML systems to develop a proof-of-concept system for visually impaired people to play chess. To this end, we developed a system that follows three major steps. First, using an audio-to-text (gpt-4o-transcribe) library and an LLM, the system converts audio input from the visually impaired user to a text representation. Second, the resulting text representation is incorporated into chess AI, Stockfish, which generates the optimal countermove. Last, the generated optimal counter-move from Stockfish is inputted into a secondary text-to-speech model used to communicate with a visually impaired user. In this study, we develop a computerized system that governs (a) person versus computer, (b) person versus person, and (c) training with puzzle modes. We also present a proof of concept demonstration through screenshots taken from the UI showing system functionality for different cases. The system developed here has the potential to bring the world of chess to visually impaired users in an accessible and intuitive way.

## 2. Materials and Methods

### 2A. Data

In this paper, we leverage data in a number of ways, from how we encode the data to how we utilize pretrained models that have had their performance optimized by large data corpora. The first data archetype we utilize is an encoding of the chess game generated by the Python Chess Library. This package is an open source set of Python functions and code that we use to generate, update, and maintain an internal representation of the state of the chess game. The second is a large language model (LLM). An LLM is a model that has been developed by scraping huge amounts of text data across the internet and various sources, which uses vector encodings of that data to make predictions about the next series of text data that comes after a given input. In our study, we prompt the LLM to predict how a given text representation of an audio input can be converted into a uniform syntax that our chess AI model is expecting and can understand. We chose to use gpt-4.1 due to its strong results in understanding text inputs, but this model could be swapped out for another as new models emerge in the future.

Third, our understanding of the chessboard and potential future moves is possible due to our use of the Stockfish chess AI model. Stockfish is an industry-leading machine learning model that has been trained on a massive dataset of past chess games. This model utilizes alpha-beta pruning and other ML techniques to predict the best chess move that a player can make to have the highest odds of winning. We leverage this model to come up with the moves that the computer uses against the player.

Fourth, to make our application accessible to users with visual impairments, we utilize an audio-to-text model (gpt-4o-transcribe) to understand the user's input. This model takes in audio data and based on frequency, amplitude, and other aspects, predicts the most accurate text representation of the audio data that is given. Finally, we use a text-to-speech model (gpt-4o-mini-tts) to communicate the computer's move back to the player. This model will generate the audio output of the computer's move. This model is a machine learning model that has been trained on a large corpus of text and audio samples in order to understand the relationship between the two and predict what the audio output should be.

### *2B. Methods*

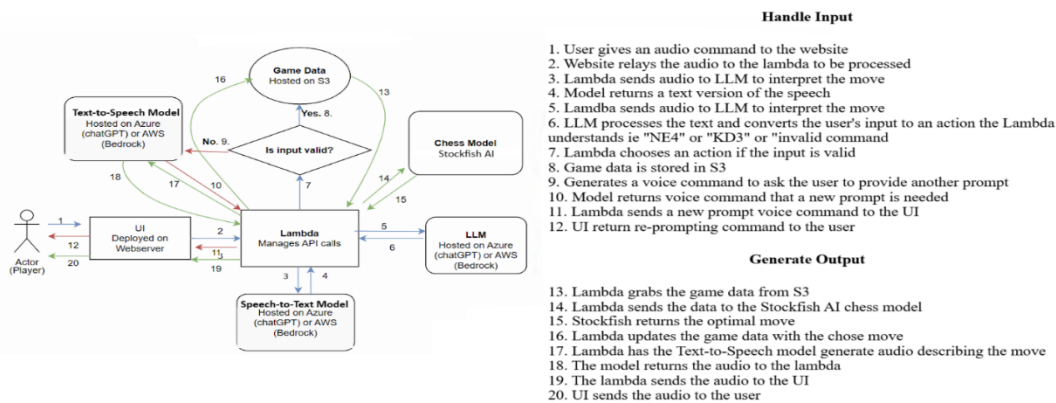
In the flow of our system, we start by taking in user input. When the user speaks into their microphone, our system captures the audio data describing the move that they want to make. From there, we use an audio-to-text ML model that has been trained to convert audio data to a text representation of that input. Then, we utilize an LLM, specifically gpt-4.1, which is produced by OpenAI, to normalize the data that the user provides. When we are working with an audio-to-text model, it doesn't always convert the data into the same format. The LLM can take the data that is passed in and convert it into a uniform format to be used throughout our system. Once we have normalized our input data, we pass it to the Stockfish chess engine, which is an AI model that has been trained to analyze chess positions and provide guidance on the best possible next move. We use this model to determine the move that the computer makes in response. As the chess game is played, we utilize the Python chess pip library to store the state of the game and also to ensure that all moves made by the chess player and computer are legal. Once the Stockfish engine has determined the optimal move to make in this situation, we pass that move to a text-to-speech model. That model converts the text data to an audio file that is played for the user.

## **3. Results**

### *3A. Model Development*

This section describes the models we built to enable visually impaired individuals to play chess. Currently, there is a renaissance in the development of machine learning models, most prominently with the recent development of large language models (LLMs) like ChatGPT, as well as many others (Hurt, 2024). While there are many models being developed constantly to solve specific use cases, the contribution of this study is to combine different models to build a cohesive system designed to help people, especially those with visual impairments. This computerized system leverages recently developed speech-to-text conversion models, LLMs, and chess AI models in a novel way to provide a chess learning experience that is not available on traditional chess websites that are designed for people with no visual impairments. As new models continue to be developed, providing greater ability for the computer to communicate with users through text or audio, future versions of this platform can grow and utilize those new models.

In this system, a chess player can play against a computer or against a physical opponent, or train with puzzles. We provide all these experiences to visually impaired individuals through an interface that combines visual and/or verbal inputs. We start by describing the system through which a blind person can play chess against the computer (Figure 1).

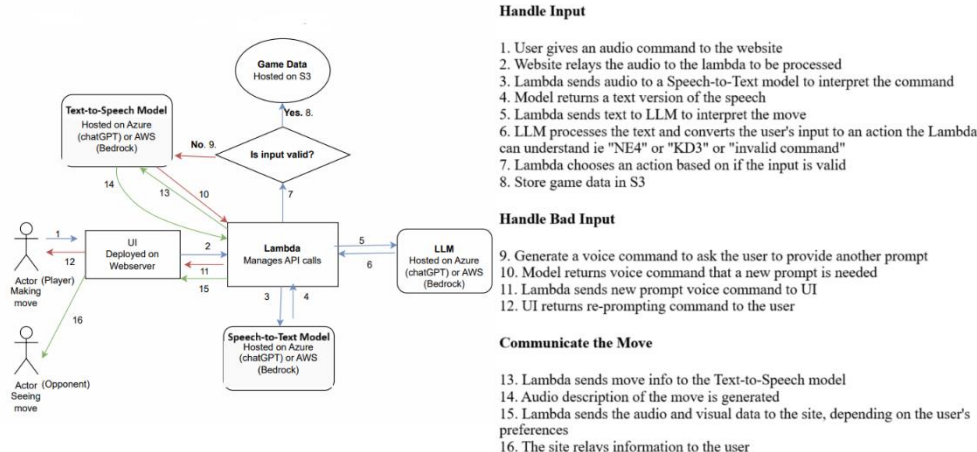


**Figure 1.** An Overview of the Conceptual Flow Chart of Steps Involved in Person vs Computer Scenario.

In Figure 1, we illustrate how we can combine several different AI and ML models to provide the experience of a person playing chess against the computer in a way that is accessible to a person with visual impairments. The left panel presents a sketch of the system sequentially, while the right panel lists the sequential steps. Steps 1-12 in Figure 1 sequentially illustrate how the player's verbal inputs are processed, and steps 13-20 sequentially depict how the computer generates a move and communicates that to the user. The actor/player in the figure above is a player with visual impairments.

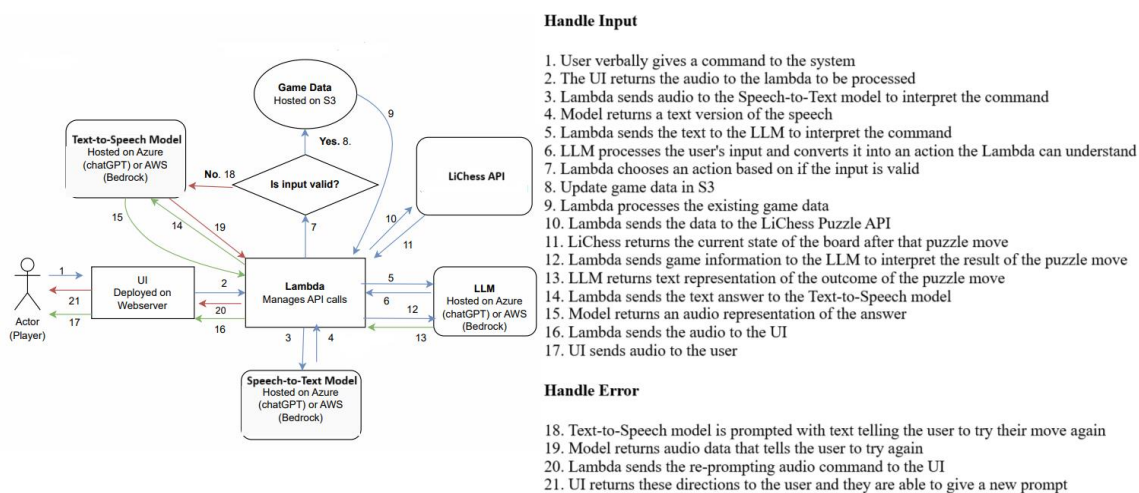
The system consists of a front-end user interface (UI) that accepts and returns both text and audio information, as well as a backend Amazon Web Services (AWS) Lambda function that is hosted in the cloud and manages the orchestration of the various AI models. The lambda uses speech-to-text and text-to-speech models to convert from an audio representation of input to a text representation of the output. That text representation is passed to the LLM to understand what the user would like to do during the game. The computer's moves are generated based on the output of an AI chess model (Stockfish) that is prompted with the current state of the board after a user's move. The computer's move will be communicated to the player via an audio output that is generated by the AI models, and a stored representation of the game in S3, an Amazon cloud storage solution, is updated after each move.

The System also provides the opportunity to play with another physical opponent, be it a blind person or a person without visual difficulty. That process is described in Figure 2. As shown in Figure 2, the visually impaired player makes a move via an audio prompt, as in Figure 1. That move will be apparent to the opponent via both an audio and text response via the UI, allowing the opponent to be visually impaired or unimpaired. When the opponent, who can be visually impaired or unimpaired, makes a move in response, the opponent sends an audio or text message to the UI. Similar to that in the "person versus computer" case, the lambda will use the speech-to-text and text-to-speech models to convert between audio and text as needed. The LLM will parse the input from the player (and the opponent) and convert it to a text string or audio representation of the move that the lambda orchestration function can understand. Any moves made will be updated in S3 and communicated to the other player. In Figure 2, steps 1-8 sequentially illustrate how the input is handled, and steps 13-16 demonstrate how the move from the opponent is communicated to the player. Additionally, steps 9-12 depict the steps the system takes if the input from the user is bad or undecipherable.



**Figure 2.** An Overview of the Conceptual Flow Chart of Steps Involved in Person vs Person.

Additionally, the System can allow the visually impaired player to play chess puzzles as well. The control flow can be seen in Figure 3. In Figure 3, we again use the speech-to-text and text-to-speech ML models to convert any audio inputs/ outputs between the site and the backend Lambda. This time, though, we call the LLM twice. The first time is aimed at understanding the user's command and normalizing the input to be something the LiChess puzzle API can process. The second time, we call the LLM with the results of the player's move on the puzzle, and the LLM generates an explanation of the new state of the board. That response is converted into an audio response by the text-to-speech AI model and conveyed to the user. In the system flow diagram, steps 1-7 cover the system's initial attempts to validate the user's input. Once it is deemed to be a valid input, steps 8-17 show how data moves through the system as the user's puzzle turn is processed. Steps 18-21 show how the system communicates with the user in case of an error or invalid command, in which case the user will need to provide a new request.



**Figure 3.** Person Training With Chess Puzzles.

### 3B. Proof-of-Concept Testing

In this section, we present screenshot depictions of the site as the user engages with our system in each of the 3 game modes: (a) Player vs Computer, (b) Player vs Player, and (c) Puzzle/Tactics. We start with the first mode, Player vs Computer.

#### 3.B.1. Player vs Computer

Here we present images from the user interface of the system when a player plays against a computer. When the user enters the UI, they receive an audio prompt asking them to set their game mode by verbally selecting an option between “computer”, “pvp” (person vs person), and “tactics”. If the user wants to play with the computer, they say into the audio input “computer” to set the game mode. Screenshot of one such instance where the player opted to play with the computer is presented in Figure 4.

## Audio Chess

(Audio Instructions): Speak one of the following to begin the game: "computer", "tactic", or "pvp". To play with the computer, say "computer". To play with another person, say "pvp" which stands for person vs person. To do chess puzzles, say "tactic". If you select "tactic", you should next specify a difficulty of "easiest", "easier", "normal", "harder", or "hardest".

▶ 0:00 / 0:26 🔊 ⋮

(Audio Instructions): Click the microphone or press 'Enter' to start recording audio

🎤 ▶ 🔊 ⋮ 00:01

Heard input: Computer

(Audio Instructions): Enter a move in chess notation. Example: "E4"

▶ 0:00 / 0:06 🔊 ⋮

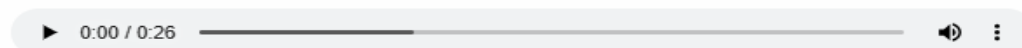


**Figure 4.** UI Screenshot of Player Choosing to Play Against Computer.

After the game mode is set, the player verbally indicates their move via the audio input. Once the player verbally makes a move, the computer, which is powered by the Stockfish chess engine, determines the best move to make in that position. It then relays it back to the user as an audio message, which they hear by using the audio player. The screenshot of this sequence is presented in Figure 5 below. In this specific instance, the player verbally communicates the move C4 while the computer returns the counter move: e7 to e5. For visual players, a depiction of the board is also shown on the screen. The game continues until either the player or the computer wins or there is a stalemate.

## Audio Chess

(Audio Instructions): Speak one of the following to begin the game: "computer", "tactic", or "pvp". To play with the computer, say "computer". To play with another person, say "pvp" which stands for person vs person. To do chess puzzles, say "tactic". If you select "tactic", you should next specify a difficulty of "easiest", "easier", "normal", "harder", or "hardest".



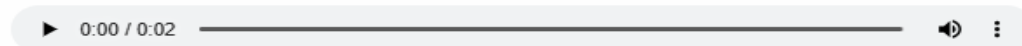
(Audio Instructions): Click the microphone or press 'Enter' to start recording audio



Heard input: C4



(Audio): Computer move e7e5



**Figure 5.** UI Screenshot of Move Sequence of Player and Computer.

### 3.B.2. Player vs Player

In Figure 6, we present screenshots from the UI for a game where the player plays against another player. When the user enters the UI for the first time, they will be verbally prompted again to select an option from the three game modes. The player says "pvp" into the audio input to activate the player vs player game mode.

## Audio Chess

(Audio Instructions): Speak one of the following to begin the game: "computer", "tactic", or "pvp". To play with the computer, say "computer". To play with another person, say "pvp" which stands for person vs person. To do chess puzzles, say "tactic". If you select "tactic", you should next specify a difficulty of "easiest", "easier", "normal", "harder", or "hardest".

▶ 0:00 / 0:26

(Audio Instructions): Click the microphone or press 'Enter' to start recording audio

▶ 00:03

Heard input: PVP

(Audio Instructions): Enter a move in chess notation. Example: "E4"

▶ 0:00 / 0:05

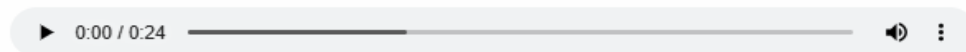


**Figure 6.** UI Screenshot where the Player Chooses to Play against another Player.

The player is assigned white or black. In this specific case, the player is assigned white. Next, the player verbally gives their move. In Figure 7 below, we see that the player verbally communicates the move "F4". The board updates to reflect the new state of the game as shown in the Figure.

## Audio Chess

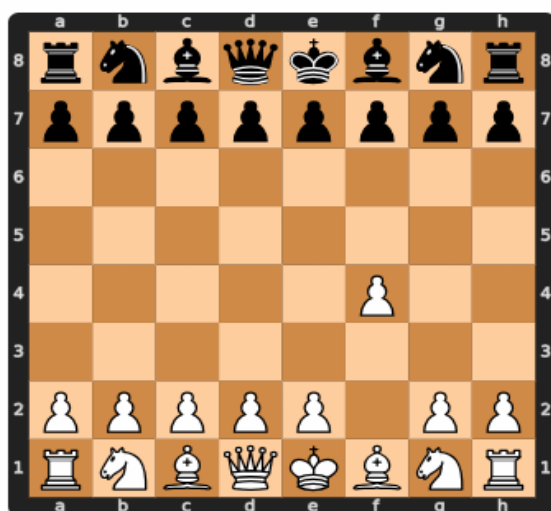
(Audio Instructions): Speak one of the following to begin the game: "computer", "tactic", or "pvp". To play with the computer, say "computer". To play with another person, say "pvp" which stands for person vs person. To do chess puzzles, say "tactic". If you select "tactic", you should next specify a difficulty of "easiest", "easier", "normal", "harder", or "hardest".



(Audio Instructions): Click the microphone or press 'Enter' to start recording audio



Heard input: F4.

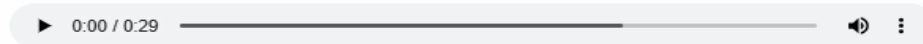


**Figure 7.** UI Screenshot of Player Verbally Making a Move in a Person Vs Person game.

After that, the opponent (black, in this case) moves by saying their move into the audio input. As seen in Figure 8, the opponent says "e7 to e5" and the board updates to show the latest sequence. The play passes back and forth between the white and black players until a winner is declared or a stalemate occurs.

## Audio Chess

(Audio Instructions): Speak one of the following to begin the game: "computer", "tactic", or "pvp". To play with the computer, say "computer". To play with another person, say "pvp" which stands for person vs person. To do chess puzzles, say "tactic". If you select "tactic", you should next specify a difficulty of "easiest", "easier", "normal", "harder", or "hardest".



(Audio Instructions): Click the microphone or press 'Enter' to start recording audio



Heard input: e7 e5.



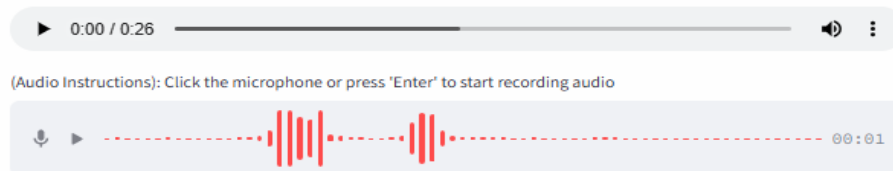
**Figure 8.** UI Screenshot after Opponent Indicates Move in a Player Vs Player game.

### 3.B.3. Chess Puzzles

In this subsection, we depict the user interface as the user prompts the system for puzzles. In Figures 9–12, we present screenshots where the user first solves a puzzle in the “easiest” category and then one in the “hardest” category. The process starts when the player says "tactic" via the audio input field. This sets the game mode to tactic as shown in Figure 9 below.

## Audio Chess

(Audio Instructions): Speak one of the following to begin the game: "computer", "tactic", or "pvp". To play with the computer, say "computer". To play with another person, say "pvp" which stands for person vs person. To do chess puzzles, say "tactic". If you select "tactic", you should next specify a difficulty of "easiest", "easier", "normal", "harder", or "hardest".



Heard input: Tactic



**Figure 9.** UI Screenshot when the User Prompts the System for Chess Puzzles.

### 3.B.3.1 Chess Puzzles: Easiest Category

Next, the user chooses one of five game modes: easiest, easier, normal, harder, or hardest. Upon specifying a game mode, the site will use the Lichess API to generate a puzzle of the requested difficulty. First, in Figure 10, the UI from a game is shown where the user has selected the "easiest" difficulty. LiChess API accordingly has created a puzzle with the "easiest" level of difficulty. As Figure 10 shows, the computer then communicates the state of the board to the player verbally as well as presents the board visually.

## Audio Chess

(Audio Instructions): Speak one of the following to begin the game: "computer", "tactic", or "pvp". To play with the computer, say "computer". To play with another person, say "pvp" which stands for person vs person. To do chess puzzles, say "tactic". If you select "tactic", you should next specify a difficulty of "easiest", "easier", "normal", "harder", or "hardest".

▶ 0:00 / 0:25

(Audio Instructions): Click the microphone or press 'Enter' to start recording audio

▶ 00:01

Heard input: Easiest.

(Audio Instructions): Enter a move in chess notation. Example: "E4"

▶ 0:00 / 0:07



(Audio): Description of the board from previously played moves

Previous moves: e4 e5 Nf3 Nc6 d4 exd4 c3 dxc3 Bc4 cxb2 Bxb2 Bb4+ Nc3 Nf6 O-O O-O e5 Ne8 Ne4 d6 exd6 Nxd6 Nxd6 Bxd6 Qc2 Bg4 Kh1 Bxf3 gxf3 Qh4 f4 Qxf4 f3 Rae8 Bd5 Nb4 Rg1 g6

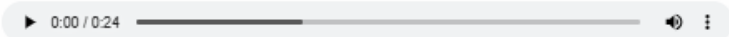
▶ 0:00 / 0:42

**Figure 10.** User Interface of Computer Generating a Puzzle with "Easiest" Difficulty.

Next, the user verbally offers their solution to the puzzle, which in this case is "g1 to g6," shown in Figure 11. Next, the system detects that the answer to the puzzle is correct and shows the new state of the board and verbally communicates to the user that their move is correct. If more moves are remaining in the puzzle, the system will also prompt the user with the computer's follow-up move.

# Audio Chess

(Audio Instructions): Speak one of the following to begin the game: "computer", "tactic", or "pvp". To play with the computer, say "computer". To play with another person, say "pvp" which stands for person vs person. To do chess puzzles, say "tactic". If you select "tactic", you should next specify a difficulty of "easiest", "easier", "normal", "harder", or "hardest".



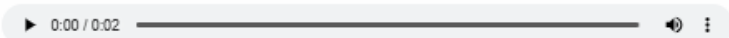
(Audio Instructions): Click the microphone or press "Enter" to start recording audio



Heard input: g1g6

g1-g6

(Audio) Correct!



(Audio): Computer move h7g6

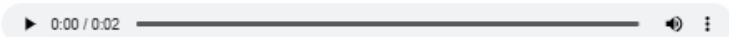


Figure 11. UI Screenshot of User's Verbal Response to the "easiest" category puzzle.

## Audio Chess

(Audio Instructions): Speak one of the following to begin the game: "computer", "tactic", or "pvp". To play with the computer, say "computer". To play with another person, say "pvp" which stands for person vs person. To do chess puzzles, say "tactic". If you select "tactic", you should next specify a difficulty of "easiest", "easier", "normal", "harder", or "hardest".

0:00 / 0:24

(Audio Instructions): Click the microphone or press 'Enter' to start recording audio

0:00 / 0:02

Heard input: h7h6

h7-h6

(Audio) Correct!

0:00 / 0:01



(Audio): Computer move d3e3

0:00 / 0:02

## Audio Chess

(Audio Instructions): Speak one of the following to begin the game: "computer", "tactic", or "pvp". To play with the computer, say "computer". To play with another person, say "pvp" which stands for person vs person. To do chess puzzles, say "tactic". If you select "tactic", you should next specify a difficulty of "easiest", "easier", "normal", "harder", or "hardest".

0:00 / 0:20

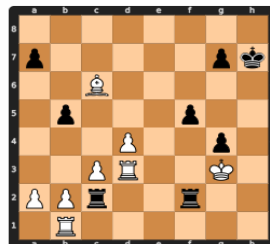
(Audio Instructions): Click the microphone or press 'Enter' to start recording audio

0:00 / 0:02

Heard input: Hardest

(Audio Instructions): Enter a move in chess notation. Example: "E4"

0:00 / 0:08



(Audio): Description of the board from previously played moves

Previous moves: e4 c5 d4 d5 Nc3 dxe4 Nxe4 Nf6 Nxf6 exf6 Nf3 Bg4 Be2 Bde O-O O-O c3 Nd7 Qd3 Re8 Bd2 Qc7 Bd1 Re7 Ng5 Bxb2+ Kh1 fng5 Bxg4 Rf4 g3 Bxd2 h6 Kg2 Rxe8 Rh1 Nf6 Bd1 Ne4 Qd3 f5 Bc2 Nvg3 Bb3+ Kh8 Qxg3 Qxg3+ Kxg3 Re2 Rab1 Kh7 Rhd1 h5 Kg2 b5 Bf7 R8e4 Bxh5 Rc2 Bf3 Rf4 Rdc1 Rd2 Rd1 Rc2 Rd3 g4 Bxc6 Rff2+ Kg3

0:00 / 1:15

**Figure 12.** UI screenshots of "hardest" difficulty puzzle.

### 3.B.3.1 Chess Puzzles: Hardest Category

Next, we present UI Screenshots where the user chooses to do a puzzle in the "hardest" category. The user verbally prompts for a puzzle in that category. This prompts the site to go to the LiChess API to retrieve a "hardest" difficulty puzzle. In the left panel of Figure 12, we can see that the site has loaded the puzzle. The user can then use the site and attempt to solve the puzzle as was described for the easiest difficulty puzzle. At the bottom of the screen is an audio player that describes the current state of the puzzle by specifying moves that have led to this state of the puzzle. When a user is ready to guess the next move of the puzzle, they can pass that move through the audio input. They will then hear a response on whether the move was correct or not.

In this specific example, the user moves "h7 to h6" as we see in the right panel in Figure 12. The computer returns an audio message that the move is correct. If there are more steps left in the puzzle, they will also hear the computer's countermove. In this case, the computer countered by moving "d3 to e3" (Figure 12, right panel). The user can then reprompt with the next move. This process continues until the puzzle is completed.

## 4. Discussion and Conclusion

Though the varied benefits of chess – intellectual, social, and psychological – are well-documented, people with disabilities still confront many barriers to enjoying the full range of these benefits. This paper is an attempt to bring chess closer to one such population – people with visual impairments. We leveraged recent advances in the application of machine learning models to develop a consistent system that has the potential to significantly improve the playing experience for such persons.

Our model builds upon large language models (LLMs), speech-to-text conversion models, and chess AI models. The proof-of-concept system developed in this research helps in making chess more accessible to visually impaired people by providing it in a new medium that is intuitive and natural to use. By combining several different models to build one cohesive solution, we are able to take chess and the cognitive and other benefits of learning it and make those more accessible than ever.

As more powerful models continue to emerge that provide a deeper ability to communicate with users and provide deeper insights into their questions and gameplay, the current platform can be updated to be even more helpful in the future. The next step involves the completion of the development of our app, which will help blind people play chess, allowing players from across the world to be matched up in online audio chess games. But these ideas have extensionality beyond the world of chess. Future work could expand on this research, whereby people with vision difficulties can become an almost equal partner to people without vision difficulties in many additional sports, beginning with other board games first. Particularly, those games similar in concept to chess can be adapted into a similar system. It may also be possible to apply machine learning and artificial intelligence to invent exciting variants of existing games. Overall, with the development of this app, we hope to create a new world where people with various visual disabilities are not at a disadvantage but are on equal footing, and where they are able to fully enjoy all of the myriad benefits of engaging in chess.

**Acknowledgments:** I am grateful to Carter Ellison and Morteza Sarmadi for mentorship, guidance, encouragement, and many insightful suggestions and discussions.

## References

1. "Accessible Chess Tutor (ACT)". University of North Carolina Department of Computer Science. Enabling Technology, COMP 290-038. Accessed August 16, 2025. <http://www.cs.unc.edu/Research/assist/et/projects/GameChest/unc>
2. Agrimi, Emanuele, Chiara Battaglini, Davide Bottari, Giorgio Gnecco and Barbara
3. Leporini. "Game accessibility for visually impaired people: a review." *Soft Computing* 28 (2024): 10475–10489, Accessed August 21, 2025.
4. <https://link.springer.com/article/10.1007/s00500-024-09827-4>
5. American Foundation for the Blind. "Statistics About Children and Youth with Vision Loss." September 2023. Updated February 2025. Accessed August 24, 2025. <https://afb.org/research-and-initiatives/statistics/children-youth-vision-loss>.
6. American Printing House. "Playing the Gambit: How the Monarch's Chess App Engages Students While Building Crucial Skills". Last modified April 19, 2024. Accessed August 16, 2025. <https://www.aph.org/blog/playing-the-gambit-how-the-monarchs-chess-app-engages-students-while-building-crucial-skills/>
7. Chase, William G. and Herbert A. Simon. "Perception in chess." *Cognitive Psychology*, Volume 4, Issue 1 (January 1973): 55-81. Accessed August 22, 2025. <https://www.sciencedirect.com/science/article/abs/pii/0010028573900042>.
8. Chess.com. "Top 10 Benefits of Chess." Last modified March 17, 2022. Accessed August 21, 2025. <https://www.chess.com/article/view/benefits-of-chess>.
9. Gnecco, Giorgio, Chiara Battaglini, Francesco Biancalani, Davide Bottari, Antonio
10. Camurri & Barbara Leporini. "Increasing Accessibility of Online Board Games to
11. Visually Impaired People via Machine Learning and Textual/Audio Feedback: The Case
12. of "Quantik"." Conference paper, International Conference on Intelligent Technologies for
13. Interactive Entertainment (INTETAIN 2023), pp 167–177, Accessed August 21, 2025.
14. [https://link.springer.com/chapter/10.1007/978-3-031-55722-4\\_12](https://link.springer.com/chapter/10.1007/978-3-031-55722-4_12)
15. Health Policy Institute. "Visual Impairments." McCourt School of Public Policy, Georgetown University, 2008. Accessed August 24, 2025. <https://hpi.georgetown.edu/visual/#:~:text=Almost%2020million%20Americans%20%E2%80%94%20people%20age%2065%20and%20older>.
16. Hurt, Brett A. "How AI is birthing a Renaissance 2.0 in the coming 'Age of a Billion Dreams'." *Medium*, April 9, 2024. Accessed August 21, 2025. <https://databrett.medium.com/how-ai-is-birthing-a-renaissance-2-0-in-the-coming-age-of-a-billion-dreams-c157c1202685>.
- 17.

18. Lichess. "How ChatGPT can Help Improve Your Chess Skills." Last modified May 26, 2023. Accessed August 24, 2025. <https://lichess.org/@/gh1234/blog/how-chatgpt-can-help-improve-your-chess-skills/6gOwZANV>
19. Merritt, Rick. "What Is Retrieval-Augmented Generation, aka RAG?" *NVIDIA blog*, Last modified November 18, 2024. Accessed August 23, 2025. <https://blogs.nvidia.com/blog/what-is-retrieval-augmented-generation/>.
20. Reznikov, Ivan. "How good is ChatGPT at playing chess?" *Medium*, Last modified December 10, 2022. Accessed August 21, 2025. <https://medium.com/@ivanreznikov/how-good-is-chatgpt-at-playing-chess-spoiler-youll-be-impressed-35b2d3ac024a>.
21. Ross, Philip E. "The Expert Mind." *Scientific American*, Vol. 295, No. 2 (August 2006): 64-71. Accessed August 20, 2025. <https://www.jstor.org/stable/26068925>.
22. Yang, Regan. "The Multifaceted Impact of Chess: Cognitive Enhancement, Educational Applications, and Technological Innovations." *Highlights in Science Engineering and Technology*, 124 (February 2024): 96-101. Accessed August 24, 2025. [https://www.researchgate.net/publication/389141577\\_The\\_Multifaceted\\_Impact\\_of\\_Chess\\_Cognitive\\_Enhancement\\_Educational\\_Applications\\_and\\_Technological\\_Innovations](https://www.researchgate.net/publication/389141577_The_Multifaceted_Impact_of_Chess_Cognitive_Enhancement_Educational_Applications_and_Technological_Innovations)

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.