Inclusive Gaming with Voice-Controlled Chess

Empowering Motor-Impaired Users Through Speech-Driven Chess

Design and Implementation of a Voice-Commanded Chess Engine for Motor-Impaired Users

Project Domain: Assistive Technology & Inclusive Gaming

Technical Stack:

Vanilla JavaScript, Web Speech API, CSS3, HTML5

I. ABSTRACT

Digital gaming often presents significant challenges for individuals with motor disabilities who struggle with traditional mouse-and-keyboard interfaces. This project introduces a voice-controlled chess engine designed for complete hands-free operation. By integrating the Web Speech API with a continuous listening loop and a robust regex-based command parser, the system translates natural language into legal chess moves. This project successfully combines algorithmic game logic with accessible human-computer interaction.

II. PROBLEM STATEMENT

Traditional digital chess platforms rely heavily on "click-and-drag" mechanics, which can be exclusionary for individuals with conditions such as quadriplegia, advanced arthritis, or muscular dystrophy. Existing voice solutions often suffer from "timeout" issues, where the microphone deactivates after brief silences. A crucial solution is a "persistent-listening" engine that remains active throughout the match duration.

III. DESIGN SPECIFICATIONS & FEATURES

A. Persistent Speech Recognition

The system utilizes a recursive on-end event trigger. When the voice status is 'Active', the system auto-re-initializes the microphone upon ending, bypassing the browser's default timeout settings.

B. Natural Language Command Parser

- **Coordinate Recognition**: Converts phrases like "D 2 to D 4" into array coordinates ([6,3]\rightarrow[4,3]).
- **Phonetic Error Correction**: Automatically recognizes homophones, converting "for" to "4" and "to" to "2".
- **Macro Phrases**: Supports high-level expressions such as "Kingside Castle" for multiple-piece moves.

C. Visual Optimization

For users with vision impairments, the application offers high contrast with solid Unicode characters, a CSS "drop-shadow" effect for visibility, and adherence to "Web Content Accessibility" guidelines.

IV. SYSTEM ARCHITECTURE

The application follows the Model-View-Controller paradigm:

- **The Model**: A 2D array representing the (8 \times 8) grid, reflecting piece positions, castling possibilities, and en passant targets.
- **The View**: A dynamic DOM visualization engine responsible for rendering square statuses, legal moves, and promotion modals.
- **The Controller**: Integrates the Speech Recognition component and event listeners to verify moves according to FIDE regulations.

V. TECHNICAL CHALLENGES & SOLUTIONS

Challenge	Solution
Microphone Timeout	Implemented a state-check in the recognition.onend listener to restart the API call if timeout occurs.
Ambiguous Input	Used a dictionary-mapping to encode number-word input into integers before regex processing.
Piece Visibility	Utilized CSS enhancements for improved visibility.

VI. SOCIAL IMPACT (EPICS FOCUS)

This project contributes to UN Sustainable Development Goal 10 (Reduced Inequalities). It enables individuals with disabilities to engage in intellectual competitions and social games through a zero-cost, browser-based assistive system, eliminating the need for expensive specialized hardware like eye trackers and puff switches.

VII. CONCLUSION & FUTURE SCOPE

The Voice-Commanded Chess Engine proves that complex games can remain accessible despite their intricacies. Future developments include:

- **Voice Feedback (TTS)**: Voice playback of opponent moves.
- **Piece-Name Recognition**: Allowing commands like "Knight to C3".
- **AI Integration**: Enabling players to play against the engine using voice commands alone.