Lex-Based Word Chain Game - Project Report

Project Title:

Lex-Based Word Chain Game with Multiple Play Modes

Introduction:

The Word Chain Game is a creative implementation designed using Lex and C as part of a Compiler Design course project. It showcases how lexical analyzers (Lex) can be extended to real-time applications beyond language parsing -- here, to handle user input parsing, state transitions, and game logic control.

Objectives:

- Utilize Lex to tokenize and process player inputs.
- Implement game states and transitions via start conditions in Lex.
- Use C functions to handle gameplay logic like word validation and turn management.
- Provide multiple play modes with dynamic behavior using mode-specific handlers.

Tools and Technologies:

- Lex (Flex): Token recognition and mode control
- C language: Game logic and state updates
- Dictionary.txt: For validating entered words
- Standard Libraries: stdio.h, stdlib.h, string.h, ctype.h, time.h, conio.h

Game Modes:

1. 2-Player Mode
2. Computer Mode
3. Timer Mode
4. Paused Mode

Integration of Lex and C:

Lex handles token matching and triggers C functions. C manages dictionary checks, game logic, and state transitions.

Code Flow Summary:

- main() - Initializes game and loads dictionary
- handle_word() - Validates and processes each word
- Mode-specific handlers - Manage turns and game flow
- Timer logic - Uses system time and sleep functions

Sample Commands:

- '2-player mode' - Starts 2-player mode
- 'computer mode' - Starts computer mode
- 'timer mode' - Starts timer mode
- 'pause', 'resume', 'end' - Game control
- <word> - Valid word input

Features:

- Multiple game modes
- Input parsing with Lex
- Dictionary-based validation
- Turn handling
- Timer mode with 20s timeout
- Pause and resume support

Team Members:

- CS22B1100 - R. Shrinidhi
- CS22B1012 - G. Satya Priya
- CS22B1017 - K. Sri Harsha Priya

- CS22B1009 - N. Priyadarshini

Conclusion:

The project successfully demonstrates the use of Lex for interactive applications, applying compiler design principles to real-world problems like input handling and state management in games.