MiniVim Project Documentation

Project Members: Yibin LOU, Yuhan RUI, Zhenhong GUO

Project Overview

MiniVim is a lightweight text editor inspired by Vim. It implements core editing functionalities and supports three modes: **Normal Mode**, **Insert Mode**, and **Command Mode**. MiniVim provides features such as **file operations, text editing, multi-file navigation, undo/redo operations, and line-based command functionalities**, fulfilling basic text editing needs. Below are the detailed descriptions of features and usage.

Modes and Features

1. Normal Mode

Purpose: This is the default mode of MiniVim, used for cursor navigation, deletion, copying, and pasting.

Feature List:

- Cursor Navigation
  - `h` and `←`: Move the cursor left.
  - `j` and `↓`: Move the cursor down.
  - `k` and `↑`: Move the cursor up.
  - `l` and `→`: Move the cursor right.
- Line Operations
  - `0`: Move the cursor to the beginning of the current line.
  - `$`: Move the cursor to the end of the current line.
  - `dd`: Delete the current line.
  - `yy`: Copy the current line.
  - `p`: Paste the copied line below the current line.
- Jump Operations
  - `gg`: Jump to the beginning of the file.
  - `G`: Jump to the end of the file.
- Undo and Redo
  - `u`: Undo the last operation.
  - `Ctrl+r`: Redo the last undo operation.
- Window Adjustment
  - When the cursor moves to the edge of the window, MiniVim automatically scrolls to keep the cursor visible.
  - Vertical Scrolling:

        - The window scrolls up if the cursor moves above the window's top edge.
        - The window scrolls down if the cursor moves below the window's bottom edge.
    - Horizontal Scrolling:
        - The window scrolls left if the cursor moves beyond the left boundary.
        - The window scrolls right if the cursor moves beyond the right boundary.

---

2. Insert Mode

Purpose: Insert text content.

Feature List:

- Cursor Navigation
    - `←`: Move the cursor left.
    - `↓`: Move the cursor down.
    - `↑`: Move the cursor up.
    - `→`: Move the cursor right.
- Enter Insert Mode: Press `i` in Normal Mode.
- Input Any Character: Insert characters at the cursor position.
- Enter Key (`Enter`): Insert a new line after the current cursor position.
- Backspace Key (`Backspace`): Delete the character before the cursor.
- Cursor Behavior:
    - The cursor can move freely to unedited areas, and spaces are automatically padded when typing.
- Exit Insert Mode: Press `Esc` to return to Normal Mode.
- Window Adjustment: Same as Normal Mode.

---

3. Command Mode

Purpose: Execute file operations, search and replace, multi-file management, and more.

Feature List:

- **Enter Command Mode**: Press `:` in Normal Mode to enter Command Mode, where commands are displayed at the bottom of the window.
- File Operations (Press `Enter` after typing the command)
    - `:w`: Save the current file.
    - `:q`: Quit the editor.
    - `:wq`: Save and quit the editor.
- Line Jump
    - Input a line number and press `Enter` (e.g., `:5`): Jump to line 5.
- Search and Replace

    - `:s/old/new/g`: Replace all occurrences of the old string with the new string in the current line.
- Multi-File Management
    - Open multiple files during initialization.
    - `:e filename`: Open or switch to the specified file.
    - `:N`: Switch to the previous file.
    - `:n`: Switch to the next file.
    - `:ls`: List currently opened files (press any key to exit).
    - `:b file_number`: Switch to the file with the specified number.

Usage Instructions

1. **Start MiniVim**:

```
# Install dependencies
sudo apt-get install libncurses5-dev
# Compile the source code into an executable file MiniVim:
g++ -o MiniVim MiniVim.cpp -lncurses
./MiniVim filename
# Example:
./MiniVim file.txt # Open a file
./MiniVim file1.txt file2.txt file3.txt  # Open multiple files simultaneously
# The mode and currently edited filename are displayed at the bottom of the
window upon startup.
```

2. **Mode Switching**:

- The default mode is **Normal Mode** after startup.
- Press `i` to switch to **Insert Mode**.
- Press `:` to switch to **Command Mode**.
- Press `Esc` to return to **Normal Mode**.

3. **Multi-File Management**:
    - Use `:e` command to open a new file, which will be added to the file history.
    - Use `:N` or `:n` to switch between files.
4. **Undo and Redo**:
    - In **Normal Mode**, press `u` to undo.
    - Use `Ctrl+r` to redo.
5. **Quit the Editor**:
    - `:q`: Quit (a prompt appears if there are unsaved changes).
    - `:wq`: Save and quit.

Design Description

This project is implemented using the **ncurses library**. When the program is run from the command line, it reads the file paths provided as arguments. After the files are read, the code will use the functions of ncurses library to initialize the window to display the file content and some instructions at the bottom of the window. Then the program uses the **getch()** function to capture user keystrokes in real time and makes corresponding changes to the cursor position and displayed content. Finally, the modified content can be saved back to the file, achieving the functionality of a basic Vim-like text editor.

Examples and Instructions

```
./MiniVim example.txt             # Start MiniVim and open the file example.txt
```

```
i                                  # Enter Insert Mode
Hello MiniVim!                     # Input content
<Enter>                            # Create a new line
This is a lightweight editor.      # Input content
<ESC>                              # Exit Insert Mode
```

```
yy                                 # Copy the first line
p                                  # Paste the copied line below
dd                                 # Delete the current line
```

```
0                                  # Move the cursor to the start of the line
l                                  # Move the cursor right by one column
i                                  # Enter Insert Mode
-                                  # Insert a character at the current cursor
position
<ESC>                              # Exit Insert Mode
```

```
:5                                 # Jump to line 5 (does nothing if the line
does not exist)
$                                  # Move the cursor to the end of the current
line
```

```
:s/editor/Editor/g                 # Replace "editor" with "Editor" in the
current line
:w                                 # Save the file
```

```
:e newfile.txt                     # Open a new file named newfile.txt
i                                  # Enter Insert Mode
This is another file.              # Input content
<ESC>                              # Exit Insert Mode
:w                                 # Save the new file
```

```
:N                                  # Switch to the previous file example.txt
:n                                  # Switch to the next file newfile.txt
:ls                                 # Display "1.example.txt\n 2.newfile.txt"
:b 1                                # Switch to the first file example.txt
```

```
u                                   # Undo the last operation
<Ctrl+r>                            # Redo the last undo operation
```

```
:wq                                 # Save the file and quit the editor
```

Development Highlights

1. **Multi-File Support**: Implements a file history, enabling quick **switching between multiple files**.
2. **Jump and Replace**: Supports **line-specific jumping** and **pattern-based replacements in the current line**.
3. **Window Adjustment**: Automatically scrolls the window to keep the cursor visible.
4. **Efficient Undo and Redo**: Manages operation history using a **stack structure**.