

Multi-Line Input Enhancement

Summary

Enhanced the chat input in TermAI to support multi-line messages and fix paste behavior. The textarea component now properly handles pasted content with newlines and allows users to compose multi-line messages.

Changes Made

1. Key Binding Updates (cmd/chat.go)

Previous Behavior

- **Enter:** Send message immediately
- No support for multi-line input
- Pasted content with newlines would be problematic

New Behavior

- **Enter:** Add a new line in the input field
- **Alt+Enter:** Send message (Option+Enter on macOS)
- **Ctrl+Enter:** Send message (works on all platforms)
- Pasted content with newlines is preserved as-is

2. Code Changes

Lines 193-228: Updated key handling logic

```
case tea.KeyEnter:
    // Check for Alt+Enter or Ctrl+Enter to send message
    if msg.Alt || strings.Contains(msg.String(), "ctrl+enter") {
        // Send message with Alt+Enter or Ctrl+Enter
        // [message sending logic]
    }
    // Regular Enter without modifiers - let textarea handle it (adds newline)
    // Fall through to default textarea behavior
```

Line 96: Updated placeholder text

```
ta.Placeholder = "Type your message... (Alt+Enter or Ctrl+Enter to send)"
```

Line 399: Updated footer with new key bindings

```
shortcuts :=
footerStyle.Render(" Alt+Enter or Ctrl+Enter to send | Enter for new line |
Ctrl+C=quit ")
```

Lines 180-186: Moved Enter key handling for suggestions into the suggestions block

- When suggestions are showing, Enter accepts the selected suggestion
- This prevents conflict with the new multi-line behavior

3. Technical Details

Textarea Configuration

- Already configured with height of 3 lines (line 100)
- Character limit: 5000 characters (line 98)
- Focus enabled by default (line 97)

Key Detection

- `msg.Alt` detects Alt key modifier (Option on macOS)
- `strings.Contains(msg.String(), "ctrl+enter")` detects Ctrl+Enter
- Regular Enter without modifiers falls through to textarea's default behavior (adds newline)

Paste Behavior

- The `textarea` component from bubbles automatically handles pasted content
- Newlines in pasted text are preserved
- All pasted content appears in the input field and is sent as a single message when Alt+Enter or Ctrl+Enter is pressed

Testing Checklist

Basic Functionality

- [x] Build succeeds without errors
- [] Application starts correctly
- [] Chat mode opens without issues

Multi-Line Input

- [] Press Enter → adds a new line (does not send)
- [] Type multiple lines of text
- [] Input area displays multiple lines correctly
- [] Input area expands to show content (up to configured height)

Send Message

- [] Alt+Enter sends the message
- [] Ctrl+Enter sends the message
- [] Multi-line messages are sent correctly
- [] Message appears in chat with formatting preserved

Paste Behavior

- [] Paste text with newlines
- [] All newlines are preserved in the input field
- [] Pasted content is treated as single input
- [] Can continue editing pasted content before sending
- [] Sending pasted multi-line content works correctly

Command Suggestions

- [] Type "/" shows command suggestions
- [] Tab/Arrow keys navigate suggestions
- [] Enter accepts selected suggestion
- [] Suggestions don't interfere with multi-line input

Edge Cases

- [] Empty input + Alt+Enter → no message sent
- [] Only whitespace + Alt+Enter → no message sent
- [] Very long multi-line messages (near 5000 char limit)
- [] Switching between single and multi-line messages
- [] UI remains responsive during streaming responses

User Experience Improvements

1. **Clearer Instructions:** Footer now clearly shows how to send messages
2. **Intuitive Behavior:** Enter for newline matches common chat applications (Slack, Discord, etc.)
3. **Cross-Platform Support:** Both Alt+Enter and Ctrl+Enter work, ensuring compatibility
4. **Better Paste Support:** Preserves formatting of pasted content
5. **Professional UX:** Allows composition of well-formatted, multi-line messages

Backward Compatibility

Breaking Changes: Yes, but minor

- Users who were accustomed to Enter sending messages will need to use Alt+Enter or Ctrl+Enter
- The new behavior is clearly indicated in the placeholder and footer
- This is a common pattern in modern chat applications

Migration:

- No code migration needed
- Users will see the new key bindings immediately in the UI
- Help text and placeholder guide users to the new behavior

Future Enhancements

Potential improvements for future iterations:

1. Configurable key bindings (allow users to choose Enter behavior)
2. Visual indicator when input contains multiple lines
3. Line count indicator for long messages
4. Syntax highlighting in the input area for code blocks
5. Auto-detection of pasted code with language hints

Build Instructions

```
cd /home/ubuntu/termal
go build -o termal
```

Version

- Modified: December 28, 2025
- TermAI Version: 1.0.0
- Bubbletea Version: Uses textarea component from bubbles package