```
import javax.swing.*;
import javax.swing.text.AbstractDocument;
import javax.swing.text.DocumentFilter;
import javax.swing.text.AttributeSet;
import javax.swing.text.BadLocationException;
import java.awt.*;

public class Main {

    private static final int GRID_SIZE = 5;  // Number of letters in the word
    private static final int NUM_GUESSES = 6;  // Number of allowed guesses
    private static final String TARGET_WORD = "APPLE";  // Target word to guess
    private static int currentAttempt = 0;  // Current guess attempt index

    public static void main(String[] args) {
        SwingUtilities.invokeLater(Main::createAndShowGUI);
    }

    private static void createAndShowGUI() {
        JFrame frame = new JFrame("Wordle");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 500);
        frame.setLayout(new BorderLayout());

        JPanel gridPanel = new JPanel();
        gridPanel.setLayout(new GridLayout(NUM_GUESSES, GRID_SIZE, 5, 5));

        JTextField[][] grid = new JTextField[NUM_GUESSES][GRID_SIZE];

        for (int i = 0; i < NUM_GUESSES; i++) {
            for (int j = 0; j < GRID_SIZE; j++) {
                grid[i][j] = new JTextField();
                grid[i][j].setHorizontalAlignment(JTextField.CENTER);
                grid[i][j].setFont(new Font("Arial", Font.BOLD, 24));

                // Restrict input to one letter
                ((AbstractDocument) grid[i][j].getDocument()).setDocumentFilter(new
DocumentFilter() {
                    @Override
                    public void replace(FilterBypass fb, int offset, int length, String text, AttributeSet
attrs) throws BadLocationException {
                        if (fb.getDocument().getLength() + text.length() - length <= 1) {
                            text = text.toUpperCase(); // Convert input to uppercase
                            super.replace(fb, offset, length, text, attrs);
```

```
                }
            }

            @Override
            public void insertString(FilterBypass fb, int offset, String text, AttributeSet attrs)
throws BadLocationException {
                if (fb.getDocument().getLength() + text.length() <= 1) {
                    text = text.toUpperCase(); // Convert input to uppercase
                    super.insertString(fb, offset, text, attrs);
                }
            }

            @Override
            public void remove(FilterBypass fb, int offset, int length) throws
BadLocationException {
                super.remove(fb, offset, length);
            }
        });

        if (i != currentAttempt) {
            grid[i][j].setEnabled(false);  // Disable rows other than the current attempt
        }

        gridPanel.add(grid[i][j]);
    }
}

JButton submitButton = new JButton("Submit Guess");
submitButton.addActionListener(e -> {
    // Collect the user's current guess
    StringBuilder guess = new StringBuilder();
    for (int i = 0; i < GRID_SIZE; i++) {
        guess.append(grid[currentAttempt][i].getText().toUpperCase());  // Collect letters from
the current row
    }

    // Check if the guess is correct
    if (guess.toString().equals(TARGET_WORD)) {
        System.out.println("CORRECT!");
        submitButton.setEnabled(false);  // Disable further submissions
    } else {
        if (currentAttempt == NUM_GUESSES - 1) {
            System.out.println("OUT OF GUESSES. The word was: " + TARGET_WORD);
            submitButton.setEnabled(false);  // Disable further submissions
```

```
            } else {
                System.out.println("INCORRECT!.");
                // Disable the current row and enable the next row
                for (int i = 0; i < GRID_SIZE; i++) {
                    grid[currentAttempt][i].setEnabled(false);
                    grid[currentAttempt + 1][i].setEnabled(true);
                }
                currentAttempt++;
            }
        }
    });

    frame.add(gridPanel, BorderLayout.CENTER);
    frame.add(submitButton, BorderLayout.SOUTH);

    frame.setVisible(true);
  }
}
```