Programming in Java

Assignment – 3



**Name :** Dharun Balaji P

**Roll Number :** 2016108

**Department :** Electronics and Instrumentation Engineering

Problem statement:

Design a typewriter game applet where the user needs to type the displayed alphabet within a time limit. The game should keep track of the score, display the high score, and provide an option to restart the game.

Description:

The typewriter game applet is a simple game where the player has to type the alphabet displayed on the screen within a specified time. The game keeps track of the score and displays the high score achieved. When the user types the correct alphabet, the score increases, and a new alphabet is displayed. If the user types the wrong alphabet, the game ends, and the user can restart the game to try again.

Concepts used:

1. **Java Applet:** The game is implemented as a Java applet, which allows interactive graphics and user input within a web browser.
2. **User Interface:** The applet uses various UI components such as labels, text fields, and buttons to display information and receive user input.
3. **Event Handling:** The applet handles keyboard events to capture user input and button clicks to respond to user actions.
4. **Random Number Generation:** The applet generates a random alphabet for each round of the game using random number generation techniques.
5. **Timer:** The applet uses a timer to track the remaining time for each round of the game. When the time is up, the game ends.
6. **Score Tracking:** The applet keeps track of the player's score and updates it accordingly when the user types the correct alphabet.
7. **Restarting the Game:** The applet allows the user to restart the game after it ends. The game resets the score, time, and target alphabet for a fresh start.
8. **High Score Tracking:** The applet maintains a high score record and updates it when the user achieves a new high score.

Program Code:

import java.applet.Applet;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.KeyAdapter;

import java.awt.event.KeyEvent;

import javax.swing.Timer;

/\*

<Applet code=TypewriterGameApplet width=500 height=250>

</Applet>

\*/

public class TypewriterGameApplet extends Applet {

    private char targetAlphabet;

    private char wrongAlphabet;

    private int score;

    private int highScore;

    private boolean gameEnded;

    private TextField userInputField;

    private Label targetLabel;

    private Label scoreLabel;

    private Label highScoreLabel;

    private Label timerLabel;

    private Button restartButton;

    private Timer timer;

    private int timeLeft;

    private String correctWord;

    public void init() {

        targetAlphabet = generateRandomAlphabet();

        wrongAlphabet = ' ';

        score = 0;

        highScore = 0;

        gameEnded = false;

        timeLeft = 60;

        correctWord = "";

        setSize(400, 250);

        setBackground(Color.WHITE);

        setFont(new Font("Arial", Font.PLAIN, 20));

        setLayout(new BorderLayout());

        targetLabel = new Label();

        targetLabel.setFont(new Font("Arial", Font.PLAIN, 20));

        targetLabel.setAlignment(Label.CENTER);

        userInputField = new TextField(10);

        userInputField.setFont(new Font("Arial", Font.PLAIN, 20));

        userInputField.addKeyListener(new KeyAdapter() {

            public void keyPressed(KeyEvent e) {

                handleKeyPress(e);

            }

        });

        scoreLabel = new Label("Score: " + score);

        scoreLabel.setFont(new Font("Arial", Font.PLAIN, 20));

        highScoreLabel = new Label("High Score: " + highScore);

        highScoreLabel.setFont(new Font("Arial", Font.PLAIN, 20));

        timerLabel = new Label("Time Left: " + timeLeft);

        timerLabel.setFont(new Font("Arial", Font.PLAIN, 20));

        restartButton = new Button("Restart");

        restartButton.setFont(new Font("Arial", Font.PLAIN, 20));

        restartButton.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent e) {

                restartGame();

            }

        });

        restartButton.setEnabled(false);

        Panel centerPanel = new Panel();

        centerPanel.setLayout(new BorderLayout());

        centerPanel.add(targetLabel, BorderLayout.CENTER);

        Panel inputPanel = new Panel();

        inputPanel.setLayout(new FlowLayout(FlowLayout.CENTER));

        inputPanel.add(userInputField);

        Panel scorePanel = new Panel();

        scorePanel.setLayout(new FlowLayout(FlowLayout.CENTER));

        scorePanel.add(scoreLabel);

        scorePanel.add(highScoreLabel);

        Panel timerPanel = new Panel();

        timerPanel.setLayout(new FlowLayout(FlowLayout.CENTER));

        timerPanel.add(timerLabel);

        Panel bottomPanel = new Panel();

        bottomPanel.setLayout(new FlowLayout(FlowLayout.CENTER));

        bottomPanel.add(scorePanel);

        bottomPanel.add(timerPanel);

        bottomPanel.add(restartButton);

        add(centerPanel, BorderLayout.CENTER);

        add(inputPanel, BorderLayout.SOUTH);

        add(bottomPanel, BorderLayout.SOUTH);

        setTargetLabel();

        startTimer();

    }

    private void setTargetLabel() {

        if (gameEnded) {

            targetLabel.setText("Oops! Game Over. The word is: " + correctWord);

        } else {

            targetLabel.setText("Type the following alphabet: " + targetAlphabet);

        }

    }

    private void handleKeyPress(KeyEvent e) {

        if (gameEnded) {

            return;

        }

        char typedChar = Character.toUpperCase(e.getKeyChar());

        if (typedChar == targetAlphabet) {

            score++;

            scoreLabel.setText("Score: " + score);

            targetAlphabet = generateRandomAlphabet();

        } else {

            gameEnded = true;

            wrongAlphabet = typedChar;

            userInputField.setEnabled(false);

            userInputField.setText("");

            restartButton.setEnabled(true);

            if (score > highScore) {

                highScore = score;

                highScoreLabel.setText("High Score: " + highScore);

            }

            correctWord = targetAlphabet + "";

            targetLabel.setText("Oops! Game Over. The word is: " + correctWord);

            timer.stop();

        }

        setTargetLabel();

    }

    private char generateRandomAlphabet() {

        return (char) ('A' + (int) (Math.random() \* 26));

    }

    private void restartGame() {

        targetAlphabet = correctWord.charAt(0);

        wrongAlphabet = ' ';

        score = 0;

        gameEnded = false;

        userInputField.setEnabled(true);

        userInputField.setText("");

        userInputField.requestFocus();

        restartButton.setEnabled(false);

        scoreLabel.setText("Score: " + score);

        setTargetLabel();

        startTimer();

    }

    private void startTimer() {

        timeLeft = 60;

        timer = new Timer(1000, new ActionListener() {

            public void actionPerformed(ActionEvent e) {

                timeLeft--;

                timerLabel.setText("Time Left: " + timeLeft);

                if (timeLeft <= 0) {

                    timer.stop();

                    gameEnded = true;

                    userInputField.setEnabled(false);

                    restartButton.setEnabled(true);

                    targetLabel.setText("Time's up! Game Over.");

                }

            }

        });

        timer.start();

    }

}

Screenshots:





