Report On

RUNAWAY CLICK GAME.

Submitted in partial fulfillment of the requirements of the Course project in Semester III of Second Year Artificial Intelligence and Data Science

by

Gautam Chaudhari(04)

Mohammed Ali Jaffari(16)

Ayush Gupta(14)

Supervisor

Mrs. Sejal D'Mello

University of Mumbai

Vidyavardhini's College of Engineering & Technology

Department of Artificial Intelligence and Data Science

(2023-24)

Vidyavardhini's College of Engineering & Technology
Department of Artificial Intelligence and Data Science

CERTIFICATE

This is to certify that the project entitled “TIC-TAC-TOE-THE GAME” is a bonafide work of " Gautam Chaudhari(04), Mohammed Ali Jaffari(16), Ayush Gupta(14) " submitted to the University of Mumbai in partial fulfillment of the requirement for the Course project in semester III of Second Year Artificial Intelligence and Data Science engineering.

Supervisor

Mrs. Sejal D’Mello

Dr. Tatwadarshi P. N.
Head of Department

Table of Contents

Pg. No

1.OVERVIEW

Title: RUNAWAY CLICK GAME

1.1 Overview:

The "Run Away Click Game" is a Java Swing application that offers an interactive gaming experience. Within the game's window, implemented as a `JFrame`, players are presented with a single button labeled "Click Me." The objective is to click on this button as it unpredictably relocates around the window. The game restricts the window's size to 800x600 pixels and disallows resizing. A black text color is set for the button to enhance visibility. The game leverages mouse event handling through a `MouseAdapter` to detect when the mouse pointer enters the button's area, at which point the button is swiftly moved to a random position within the window, courtesy of the `moveButtonAway` method. The game's simple mechanics and visual appeal make it a fun exercise in reflexes and aiming skills, while also serving as a demonstration of Java Swing components and event-driven programming. Players launch the game through the `main` method, ensuring proper Swing operation by using `SwingUtilities.invokeLater`.

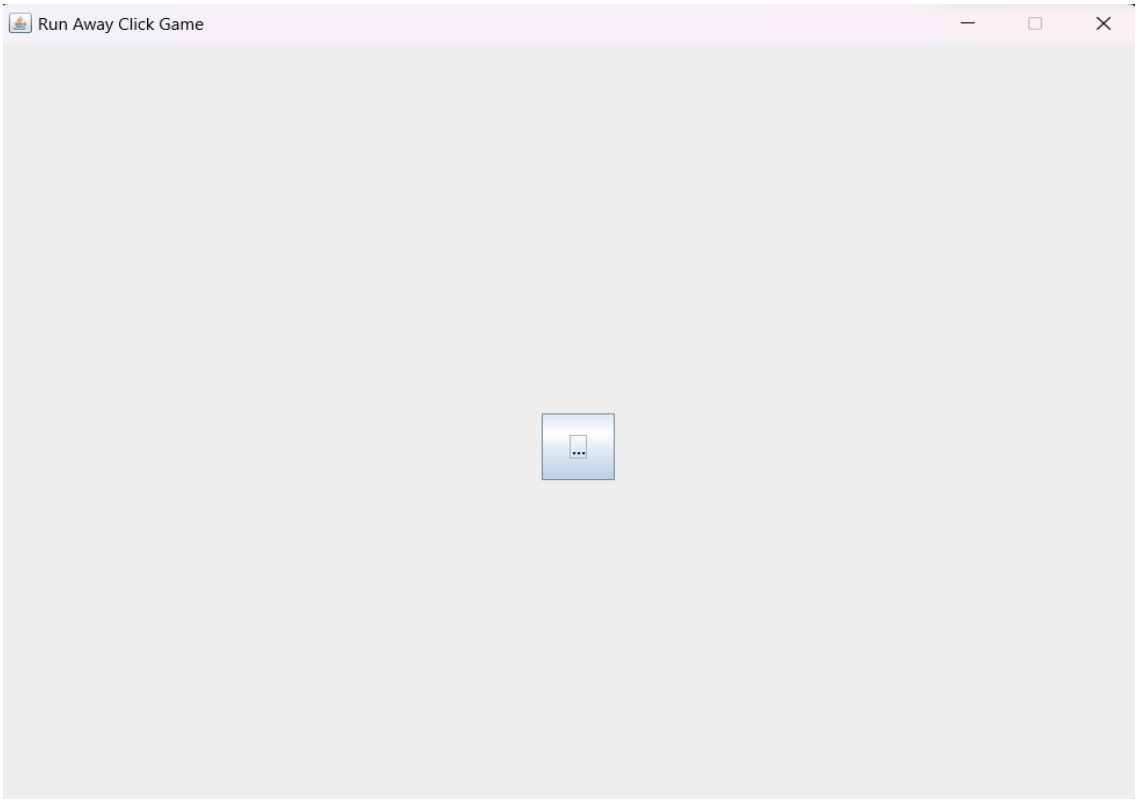
1.2 HOW TO PLAY: -

Playing the "Run Away Click Game" is straightforward and requires you to interact with the graphical user interface provided by the application. Here are the steps to play the game:

1. Launch the Game:
- Run the Java program to launch the game. The game window titled "Run Away Click Game" will appear on your screen.
2. Target the Button:
- Within the game window, you will see a button. Your objective is to click on this button when it appears.
3. Observe the Button's Movement:
- As you move your mouse pointer over the button, the button will unpredictably move to a new location within the game window.
4. React Swiftly:
- Be ready to click the button when it moves to its new location. The challenge lies in clicking the button while it is in motion.
5. Keep Clicking:
- The game's aim is to test your reflexes and accuracy. Continue clicking the button each time it changes its position.
6. Scoring:
- The game doesn't have a formal scoring system. You can challenge yourself to see how many times you can successfully click the moving button within a set time frame.
7. Enjoy the Challenge:
- The game provides a fun and engaging experience, challenging your mouse-handling skills and speed.
8. Exit the Game:
- To exit the game, simply close the game window by clicking the close button (X) in the top-right corner of the window. The game will terminate.

The "Run Away Click Game" is a simple, casual game that you can play for a quick diversion or to challenge your reflexes. There are no levels or complex rules—just the simple goal of clicking the moving button. Have fun and see how many times you can successfully click it!

1.3 SNEAK – PEEK:



2.PROGRAM:-

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
```

```
import java.util.Random;


public class RunAwayClickGame extends JFrame {
    private JButton clickButton;
    private int buttonSize = 50;
    private Random random;
    private int buttonX, buttonY;


    public RunAwayClickGame() {
        setTitle("Run Away Click Game");
        setLayout(null);
        setResizable(false);
        setSize(800, 600);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);


        clickButton = new JButton("Click Me");
        clickButton.setBounds(375, 275, buttonSize, buttonSize);
        clickButton.setForeground(Color.BLACK); // Set text color to black
        random = new Random();


        add(clickButton);


        clickButton.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseEntered(MouseEvent e) {
                moveButtonAway();
            }
        });


        setVisible(true);
    }


    private void moveButtonAway() {
        int maxWidth = getWidth() - buttonSize;
        int maxHeight = getHeight() - buttonSize;


        buttonX = random.nextInt(maxWidth);
```

```
        buttonY = random.nextInt(maxHeight);


        clickButton.setBounds(buttonX, buttonY, buttonSize, buttonSize);
    }


    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new RunAwayClickGame());
    }
}
```

3.TECHNICALITIES

3.1 TECHNOLOGIES USED:

The "Run Away Click Game" is a relatively simple game built using Java Swing, a technology for creating graphical user interfaces (GUIs) in Java applications. Here's an overview of the key technologies used in this game:

1. Java Swing: Java Swing is a set of libraries and tools for building graphical user interfaces in Java applications. It provides a wide range of components, including buttons, windows, and event handling mechanisms, which were used to create the game's graphical interface.
2. JFrame: The game's main window is implemented using a JFrame, which is a Swing component that provides the basic frame for the application. It allows for the creation of windows, dialog boxes, and other top-level containers.
3. JButton: The primary interactive element in the game is a JButton. This Swing component is used to create the clickable target that players need to interact with. It supports event handling and customization options.
4. Mouse Events: The game relies on mouse events to detect when the player's cursor enters the button's area. This is achieved using the MouseAdapter class and the mouseEntered event, which triggers the button's movement.
5. Randomization: To make the game challenging, randomization is used. The Random class from Java's java.util package is employed to generate random X and Y coordinates for the button, causing it to move to new positions within the window when the mouse pointer enters its area.
6. UI Customization: While not heavily emphasized in this game, Java Swing allows for extensive customization of the user interface. In this

case, the text color of the button is customized to black, demonstrating the ability to adjust the visual aspects of the interface.

7. Main Method and SwingUtilities.invokeLater: The game's main method uses the SwingUtilities.invokeLater method to ensure that the Swing components are created and accessed on the Event Dispatch Thread (EDT). This is crucial for the smooth operation of Swing-based applications.
8. Java Programming Language: The entire game is written in Java, a widely used and platform-independent programming language. This means that the game can be run on any system with a Java Virtual Machine (JVM) without modification.

3.2 EXPLANATION:-

1. The code begins by importing the necessary Java libraries, including Swing for GUI components and event handling, AWT for basic GUI functionality, and Random for generating random numbers.
2. The RunAwayClickGame class extends JFrame, which is a Swing component for creating the main application window.
3. In the constructor (RunAwayClickGame), the game's main window is set up with the title "Run Away Click Game." The layout is set to null, meaning components are manually positioned, and the window is fixed in size at 800x600 pixels. The application is configured to exit when the window is closed.
4. A JButton named clickButton is created, with the label "Click Me." Its initial position and size are set, and its text color is customized to black.
5. An instance of Random is created to generate random numbers for button movement.
6. The clickButton is added to the window using the add method.
7. A mouse event listener is added to the clickButton. When the mouse enters the button's area, the mouseEntered method is triggered, calling the moveButtonAway method.
8. The moveButtonAway method calculates a new random position for the button within the window, ensuring it remains visible, and then updates the button's location using setBounds.
9. The main method serves as the entry point for the game. It uses SwingUtilities.invokeLater to ensure the game's GUI components are created and accessed on the Event Dispatch Thread (EDT) to maintain Swing's thread-safety.
10. When you run the program, an instance of RunAwayClickGame is created, launching the game and displaying the main window.

11. In summary, this code creates a simple game in which a button moves randomly within a window, and the player's goal is to click on it as it changes position. The game uses Java Swing for its GUI components, AWT for basic functionality, and event handling to make the button interactive. The game provides a fun and straightforward example of a Java Swing application.

4. CONCLUSION:

In summary, the "Run Away Click Game" is a straightforward and entertaining Java Swing application that tests players' mouse-handling abilities. It's an excellent demonstration of creating a graphical user interface, event handling, and randomization for simple game mechanics. While it lacks

complex features, the game's charm lies in its accessibility and quick gameplay, making it a great choice for those looking to practice Java Swing and basic game development.