import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

→ এখানে আমরা Java Swing GUI তৈরি করতে প্রয়োজনীয় লাইব্রেরিগুলো ইমপোর্ট করছি।

javax.swing.\* – Swing কম্পোনেন্ট (যেমন: JFrame, JButton, JLabel) ব্যবহার করতে।

java.awt.\* – গ্রাফিক্স ও লেআউটের জন্য।

java.awt.event.\* – ইভেন্ট হ্যান্ডলিংয়ের জন্য (যেমন: বাটন ক্লিক)।

public class TicTacToeGUI\_V2 extends JFrame implements ActionListener {

→ আমরা একটি ক্লাস তৈরি করছি TicTacToeGUI\_V2 নামে, যা JFrame (একটি উইন্ডো) থেকে extends করছে এবং ActionListener ইন্টারফেস ইমপ্লিমেন্ট করছে — যাতে বাটন ক্লিক করলে প্রতিক্রিয়া জানাতে পারে।

private JButton[] buttons = new JButton[9];

private String turn = "X";

private JLabel statusLabel;

→

buttons – ৯টি JButton-এর অ্যারে, যেগুলো খেলার বোর্ডে ব্যবহৃত হবে।

turn – কোন প্লেয়ারের টার্ন সেটা রাখা হবে (X বা O)।

statusLabel – উপরে দেখানো হবে কে খেলছে বা কে জিতেছে।

public TicTacToeGUI\_V2() {

→ Constructor, যখন ক্লাস অবজেক্ট তৈরি হবে তখন এই অংশ চালু হবে।

setTitle("Tic Tac Toe");

setSize(450, 500);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new BorderLayout());

getContentPane().setBackground(new Color(240, 240, 240));

→

উইন্ডোর টাইটেল ও আকার ঠিক করছি।

বন্ধ করার সময় প্রোগ্রাম পুরোপুরি বন্ধ হবে।

BorderLayout ব্যবহার করছি লেআউট ম্যানেজারের জন্য।

ব্যাকগ্রাউন্ড হালকা ধূসর রঙে করা হয়েছে।

JPanel panel = new JPanel();

panel.setLayout(new GridLayout(3, 3, 10, 10));

panel.setBackground(new Color(240, 240, 240));

panel.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));

→

বোর্ড তৈরির জন্য একটি JPanel নিচ্ছি।

এতে ৩x৩ গ্রিড লেআউট ব্যবহার করছি, মানে ৯টি ঘর।

কিছু প্যাডিং ও ব্যাকগ্রাউন্ড সেট করছি।

Font font = new Font("Comic Sans MS", Font.BOLD, 60);

→ বাটনে লেখার জন্য বড়, সুন্দর ফন্ট তৈরি করছি।

for (int i = 0; i < 9; i++) {

buttons[i] = new JButton();

buttons[i].setFont(font);

buttons[i].setFocusPainted(false);

buttons[i].setBackground(Color.WHITE);

buttons[i].setForeground(new Color(50, 50, 50));

buttons[i].addActionListener(this);

panel.add(buttons[i]);

}

→ এই লুপে ৯টি বাটন তৈরি করা হচ্ছে:

ফন্ট ও রঙ সেট করা হচ্ছে।

ইভেন্ট লিসেনার যুক্ত করা হচ্ছে (বাটনে ক্লিক হলে actionPerformed কল হবে)।

panel-এ প্রতিটি বাটন যোগ করা হচ্ছে।

statusLabel = new JLabel("X's Turn", SwingConstants.CENTER);

statusLabel.setFont(new Font("Comic Sans MS", Font.BOLD, 25));

statusLabel.setForeground(new Color(60, 60, 60));

statusLabel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

→ উপরের অংশে দেখানোর জন্য একটি লেবেল তৈরি করছি — কে এখন খেলছে সেটা দেখাবে।

add(statusLabel, BorderLayout.NORTH);

add(panel, BorderLayout.CENTER);

setVisible(true);

}

→ লেবেল উপরে এবং বোর্ড মাঝখানে বসানো হচ্ছে। এরপর উইন্ডো দৃশ্যমান করা হচ্ছে।

বাটন ক্লিক হ্যান্ডলিং:

@Override

public void actionPerformed(ActionEvent e) {

JButton clickedButton = (JButton) e.getSource();

→ যদি কোনো বাটনে ক্লিক করা হয়, তাহলে সেটা কোন বাটন ছিল তা পাওয়া যাচ্ছে।

if (!clickedButton.getText().equals("")) {

return;

}

→ যদি বাটনে আগেই কিছু লেখা থাকে (মানে ওই ঘরে আগে খেলা হয়েছে), তাহলে কিছু না করে ফিরে যান।

clickedButton.setText(turn);

clickedButton.setForeground(turn.equals("X") ? new Color(30, 144, 255) : new Color(220, 20, 60));

→ এখনকার প্লেয়ারের সাইন বসানো হচ্ছে (X বা O), এবং সাদা-ব্লু-রেড রঙে সেট করা হচ্ছে।

int[] winningLine = checkWinner();

→ এখন চেক করা হচ্ছে কেউ জিতেছে কি না।

if (winningLine != null) {

highlightWinner(winningLine);

statusLabel.setText(turn + " Wins!");

JOptionPane.showMessageDialog(this, "Congratulations! " + turn + " wins!", "Game Over", JOptionPane.INFORMATION\_MESSAGE);

resetGame();

→ যদি কেউ জিতে যায়:

জেতা ঘরগুলো সবুজ করে দেওয়া হয়,

জেতার মেসেজ দেখানো হয়,

তারপর খেলা রিসেট করা হয়।

} else if (isDraw()) {

statusLabel.setText("It's a Draw!");

JOptionPane.showMessageDialog(this, "It's a Draw!", "Game Over", JOptionPane.INFORMATION\_MESSAGE);

resetGame();

→ যদি সব ঘর ভর্তি হয় কিন্তু কেউ না জেতে, তাহলে ড্র ঘোষণা করা হয়।

} else {

turn = (turn.equals("X")) ? "O" : "X";

statusLabel.setText(turn + "'s Turn");

}

→ যদি না কেউ জেতে বা ড্র হয়, তাহলে টার্ন পাল্টে দেওয়া হয়।

জয়ী চেক করা:

private int[] checkWinner() {

int[][] winConditions = {

{0,1,2}, {3,4,5}, {6,7,8},

{0,3,6}, {1,4,7}, {2,5,8},

{0,4,8}, {2,4,6}

};

→ এখানে সব জেতার সম্ভাব্য অবস্থান রাখা হয়েছে (রো, কলাম, ডায়াগোনাল)।

for (int[] condition : winConditions) {

if (buttons[condition[0]].getText().equals(turn) &&

buttons[condition[1]].getText().equals(turn) &&

buttons[condition[2]].getText().equals(turn)) {

return condition;

}

}

return null;

}

→ যদি যেকোনো তিনটি ঘরে একই প্লেয়ারের চিহ্ন থাকে, সেটা জয় — জয়ী অবস্থান রিটার্ন করা হয়।

ড্র চেক:

private boolean isDraw() {

for (JButton button : buttons) {

if (button.getText().equals("")) {

return false;

}

}

return true;

}

→ সব ঘর ভরা এবং কেউ না জিতলে ড্র।

জয়ী হাইলাইট:

private void highlightWinner(int[] winCondition) {

for (int index : winCondition) {

buttons[index].setBackground(new Color(144, 238, 144));

}

}

→ জেতা ৩টি ঘরের ব্যাকগ্রাউন্ড সবুজ করা হচ্ছে।

খেলা রিসেট:

private void resetGame() {

Timer timer = new Timer(1500, new ActionListener() {

public void actionPerformed(ActionEvent evt) {

for (JButton button : buttons) {

button.setText("");

button.setBackground(Color.WHITE);

}

turn = "X";

statusLabel.setText("X's Turn");

}

});

timer.setRepeats(false);

timer.start();

}

→ ১.৫ সেকেন্ড পরে বোর্ড ক্লিয়ার হয়, সব বাটনের টেক্সট ও রঙ রিসেট হয়, এবং আবার X থেকে শুরু হয়।

public static void main(String[] args) {

new TicTacToeGUI\_V2();

}

}

→ মূল মেথড — যেখানে প্রোগ্রাম চালু হয় এবং GUI তৈরি হয়।