1 **import** java.awt.\*;

2 **import** java.applet.\*;

3 **import** javax.swing.\*;

4 **import** java.util.Random;

5 **import** java.awt.event.\*;

6

7 **public** **class** GuessingGame **extends** JApplet **implements** ActionListener

8 {

9 **boolean** firstClick = **true**;

10 **int** firstClickValue = 0;

11 **int** firstClickIndex = 0;

12 **int** secondClickIndex = 0;

13 **int** errorCount = 0;

14 **boolean** correct = **false**;

15 **int**[] values = **new** **int**[16];

16 JButton b = **new** JButton();

17 JButton[]cell;

18 JLabel errors = **new** JLabel("Errors: ");

19 JLabel numberOfErrors = **new** JLabel("0");

20 JButton begin = **new** JButton("Begin");

21 JButton reset = **new** JButton("Reset");

22 Timer timer1 = **new** Timer(2000, **new** TimerHandler());

23

24 **public** **void** init()

25 {

26 cell = **new** JButton[16]; **//the number buttons**

27

28 setLayout(**new** GridLayout(5, 4)); **//override default BorderLayout**

29

30 **//set properties of GUI compoments**

31 errors.setHorizontalAlignment(JLabel.RIGHT);

32 errors.setFont(**new** Font("Serif", Font.BOLD, 30));

33 numberOfErrors.setFont(**new** Font("Serif", Font.BOLD, 30));

34

35 **//add event handlers to listener lists**

36 begin.addActionListener(**new** BeginResetHandler());

37 reset.addActionListener(**new** BeginResetHandler());

38

39 **//create number buttons, set properties, register event handlers**

40 **for**(**int** i = 0; i < 16; i++)

41 {

42 cell[i] = **new** JButton("0");

43 cell[i].setFont(**new** Font("Serif", Font.BOLD, 40));

44 cell[i].addActionListener(**this**);

45 add(cell[i]);

46 }

47

48 **//add the lower buttons and labels**

49 add(begin);

50 add(errors);

51 add(numberOfErrors);

52 add(reset);

53

54 intializeGame();

55 }

56

57 **public** **void** intializeGame()

58 {

59 generateNumbers();

60 numberOfErrors.setText("0");

61 errorCount = 0;

62 **for**(**int** i = 0; i < 16; i++)

63 {

64 cell[i].setText(Integer.toString(values[i]));

65 cell[i].setForeground(Color.BLACK);

66 }

67 }

68

69 **public** **void** generateNumbers() **//generates buttons' numbers**

70 {

71 Random rn = **new** Random();

72 **int** number = 0;

73 **int** cellNumber = 0;

74 **boolean** done;

75

76 **for**(**int** i = 1; i <= 16; i++) **//all number buttons**

77 {

78 values[i-1] = i % 8 + 1; **//place numbers 1->8 twice**

79 }

80 **for**(**int** i = 0; i < 16; i++) **//all number buttons**

81 {

82 number = rn.nextInt(15);

83 swap(values, i, number); **//swap button value with a random button**

84 }

85 }

86

87 **public** **void** swap(**int**[] array, **int** indx1, **int** indx2)

88 {

89 **int** temp;

90 temp = array[indx1];

91 array[indx1] = array[indx2];

92 array[indx2] = temp;

93 }

94

95 **//event handlers \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

96 **public** **class** BeginResetHandler **implements** ActionListener

97 {

98 **public** **void** actionPerformed(ActionEvent e)

99 {

100 **if**(e.getSource() == begin) **// start the game**

101 {

102 **for**(**int** i = 0; i < 16; i++)

103 {

104 cell[i].setText(" "); **//hide the numbers**

105 }

106 }

107 **else** **//generate and a new game**

108 {

109 intializeGame();

110 }

111 }

112 }

113

114 **public** **void** actionPerformed(ActionEvent e) **//number buttons' handler**

115 {

116 **if**(firstClick) //show the number

117 {

118 **for**(**int** i = 0; i<16; i++) **//all number buttons**

119 {

120 **if**(e.getSource() == cell[i]) **//button clicked found**

121 {

122 cell[i].setText(Integer.toString(values[i]));

123 firstClick = **false**;

124 firstClickValue = values[i];

125 firstClickIndex = i;

126 **break**;

127 }

128 }

129 }

130 **else** **//second click processing**

131 {

132 timer1.start(); //two seconds

133 **for**(**int** i = 0; i<16; i++) **//all number buttons**

134 {

135 **if**(e.getSource() == cell[i]) **//button clicked found**

136 {

137 cell[i].setText(Integer.toString(values[i]));

138 firstClick = **true**;

139 secondClickIndex = i;

140 **if**(firstClickValue == values[i]) **//correct match**

141 {

142 correct = **true**;

143 cell[firstClickIndex].setForeground(Color.BLUE);

144 cell[secondClickIndex].setForeground(Color.BLUE);

145 }

146 **else** **//incorrect match**

147 {

148 correct = **false**;

149 errorCount++;

150 numberOfErrors.setText(Integer.toString(errorCount));

151 }

152 **break**;

153 }

154 }

155 }

156 }

157

158 **public** **class** TimerHandler **implements** ActionListener

159 {

160 **public** **void** actionPerformed(ActionEvent e) **//Timer's event handler**

161 {

162 **if**(correct == **false**) **//no match**

163 {

164 timer1.stop(); **//after a two second pause**

165 cell[firstClickIndex].setText(" "); **//hide the numbers**

166 cell[secondClickIndex].setText(" ");

167 }

168 }

169 }

170 }

**Figure 11.47 The applet GuessingGame.**