1 **import** edu.sjcny.gpv1.\*;

2 **import** java.awt.\*;

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

4

5 **public** **class** DecisionsControlOfFlow **extends** DrawableAdapter

6 {

7 **static** DecisionsControlOfFlow ge = **new** DecisionsControlOfFlow();

8 **static** GameBoard gb = **new** GameBoard(ge, "Control Of Flow");

9 **static** BoxedSnowman s1 = **new** BoxedSnowman(300, 200, Color.GREEN);

10 **static** BoxedSnowman s2 = **new** BoxedSnowman(30, 100, Color.BLACK);

11 **static** **int** score = 0;

12 **static** **int** count = 10;

13

14 **public** **static** **void** main(String[] args)

15 {

16 showGameBoard(gb);

17 }

18

19 **public** **void** draw(Graphics g) **//call back method**

20 {

21 **int** w = 40;

22 **int** h = 77;

23 **int** s1X, s1Y, s2X, s2Y, temp;

24

25 s1X = s1.getX(); s1Y = s1.getY();

26 s2X = s2.getX(); s2Y = s2.getY();

27 g.setColor(Color.BLACK);

28 g.setFont(new Font("Arial", Font.BOLD, 18));

29 g.drawString("Time remaining: " + count, 260, 50);

30

31 **if**(count == 0) **//game over**

32 {

33 g.setColor(Color.BLACK);

34 g.drawString("Game Over", 205, 70);

35 g.drawString("Have a Good Day", 175, 90);

36 }

37 **else** **if**( !(s2X > s1X + w || s2X + w < s1X || s2Y > s1Y + h ||

38 s2Y + h < s1Y) && s1.getVisible() == **true**) **// collision**

39 {

40 score = score + 1;

41 s1.setVisible(false);

42 }

43 **else** **if**( s2X > s1X + w || s2X + w < s1X || s2Y > s1Y + h ||

44 s2Y + h < s1Y) **// no collision**

45 {

46 **if**(s1.getVisible() == false) **// not visible**

47 { temp = s1.getX();

48 s1.setX(s1.getY());

49 s1.setY(temp);

50 s1.setVisible(true);

51 }

52 }

53

54 s2.show(g);

55 **if**(s1.getVisible() == true)

56 {

57 s1.show(g);

58 }

59 g.setColor(Color.BLACK);

60 g.drawString("Score: " + score, 150, 50);

61 }

62

63 **public** **void** keyStruck(**char** key) **// call back method**

64 {

65 **int** newX, newY;

66

67 **switch** (key)

68 {

69 **case** 'L':

70 {

71 newX = s2.getX() - 2;

72 s2.setX(newX);

73 **break**;

74 }

75 **case** 'R':

76 {

77 newX = s2.getX() + 2;

78 s2.setX(newX);

79 **break**;

80 }

81 **case** 'U':

82 {

83 newY = s2.getY() - 2;

84 s2.setY(newY);

85 **break**;

86 }

87 **case** 'D':

88 {

89 newY = s2.getY() + 2;

90 s2.setY(newY);

91 }

92 } **// end of switch statement**

93 }

94 **public** **void** timer1() **// call back method**

95 {

96 count = count - 1;

97 **if**(count == 0)

98 {

99 gb.stopTimer(1);

100 }

101 }

102 }

**Figure 4.13 The DecisionsControlOfFlow application: A decision statement case study.**