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

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

3

4 **public** **class** DeepAndShallow **extends** DrawableAdapter

5 { **static** DeepAndShallow ge = **new** DeepAndShallow ();

6 **static** GameBoard gb = **new** GameBoard(ge, "Deep and Shallow Data");

7 **static** ParentSnowmanV2[] ps;

8 **static** boolean gameOver = **false**;

9

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

11 { ps = **new** ParentSnowmanV2[10];

12 ps[0] = **new** ParentSnowmanV2(100, 200, "G", Color.BLUE);

13 ps[1] = **new** ParentSnowmanV2(300, 275, "1", Color.GREEN);

14 ps[2] = **new** ParentSnowmanV2(300, 400, "2", Color.GREEN);

15 ps[3] = **new** ParentSnowmanV2(100, 100, "3", Color.GREEN);

16

17 gb.setTimerInterval(3, 20);

18 showGameBoard(gb);

19 }

20

21 **public** **void** draw(Graphics g)

22 {

23 **for**(**int** i = 1; i < ps.length; i++)

25 {

26 **if**(ps[i] != **null**) **//the snowman exists**

27 {

29 ps[i].show(g);

30 }

31 }

32 ps[0].show(g); **//the patrolling guard**

35 }

36

37 **public** **void** timer3()

38 { **int** x, speed, y;

39 **if**(ParentSnowmanV2.getSnowmanCount()== 10)

40 {

41 gb.stopTimer(3);

42 gameOver = **true**;

43 }

44 **//move the guard**

45 x = ps[0].getX();

46 x = x + ps[0].getXSpeed();

47 ps[0].setX(x);

48 y = ps[0].getY();

49 y = y + ps[0].getYSpeed();

50 ps[0].setY(y);

51

52 **//is ps[0] at a border?**

53 if(ps[0].getX() >= 460 || ps[0].getX() <= 6)

54 {

55 speed = ps[0].getXSpeed();

56 speed = -speed;

57 ps[0].setXSpeed(speed);

58 ps[ParentSnowmanV2.getSnowmanCount()] = ps[0].partialClone();

59 }

60 **if**(ps[0].getY() >= 423 || ps[0].getY() <= 30)

61 {

62 speed = ps[0].getYSpeed();

63 speed = -speed;

64 ps[0].setYSpeed(speed);

65 ps[ParentSnowmanV2.getSnowmanCount()] = ps[0].partialClone();

66 }

67

68 **// has ps[0] found a green-hat wandering snowman?**

69 **for**(**int** i = 1; i <= ps.length; i++)

70 {

71 **if**(ps[i] != null && ps[0].collidedWith(ps[i]) &&

72 !ps[0].equals(ps[i]))

73 {

74 ps[i].setX(ps[0].getX()); **//position the wanderer behind ps[0]**

75 ps[i].setY(ps[0].getY());

77 }

78 }

79 }

80

81 **public** **void** leftButton()

82 {

83 **if**(gameOver == true)

84 { **for**(**int** i=0; i<=3; i++) **//move the three intruders left**

85 {

86 ps[i].setX(ps[i].getX() – (i \* 3));

87 }

88 ps[0].setX(ps[0].getX() - 1); **// move the guard**

89 }

90 }

91 }

**Figure 7.13 The application DeepAndShallow.**