

## Chapter 9 Worksheet

```
public class AntHill extends Actor {
    private int ants = 0;
    private int maxAnts = 40;
    private Counter foodCounter;

    public AntHill() { }

    public AntHill(int numberOfAnts) {
        maxAnts = numberOfAnts;
    }

    public void act() {
        if(ants < maxAnts) {
            if(Greenfoot.getRandomNumber(100) < 10) {
                getWorld().addObject(new Ant(this), getX(), getY());
                ants++;
            }
        }
    }

    public void countFood() {
        if(foodCounter == null) {
            foodCounter = new Counter("Food: ");
            int x = getX();
            int y = getY() + getImage().getWidth()/2 + 8;

            getWorld().addObject(foodCounter, x, y);
        }
        foodCounter.increment();
    }
}
*****

public class Creature extends Actor {
    private static final int SPEED = 3;
    private int deltaX;
    private int deltaY;
    private AntHill home;

    public Creature() {
        deltaX = 0;
        deltaY = 0;
    }

    public void setHomeHill(AntHill homeHill) { home = homeHill; }

    public AntHill getHomeHill() { return home; }

    public void randomWalk() {
        if (randomChance(50)) {
            deltaX = adjustSpeed(deltaX);
            deltaY = adjustSpeed(deltaY);
        }
        walk();
    }

    public void walkTowardsHome() {
        if(home == null) { return; }
        if (randomChance(2)) { randomWalk(); }
        else {
```

## Chapter 9 Worksheet

```
    headRoughlyTowards(home);
    walk();
}
}

public void walkAwayFromHome() {
    if(home == null) { return; }
    if (randomChance(2)) { randomWalk(); }
    else {
        headRoughlyTowards(home); // first head towards home...
        deltaX = -deltaX; // ...then turn 180 degrees
        deltaY = -deltaY;
        walk();
    }
}

public void headTowards(Actor target) {
    deltaX = capSpeed(target.getX() - getX());
    deltaY = capSpeed(target.getY() - getY());
}

public void walk() {
    setLocation(getX() + deltaX, getY() + deltaY);
    setRotation((int) (180 * Math.atan2(deltaY, deltaX) / Math.PI));
}

private void headRoughlyTowards(Actor target) {
    int distanceX = Math.abs(getX() - target.getX());
    int distanceY = Math.abs(getY() - target.getY());
    boolean moveX = (distanceX > 0) && (Greenfoot.getRandomNumber(distanceX + distanceY) < distanceX);
    boolean moveY = (distanceY > 0) && (Greenfoot.getRandomNumber(distanceX + distanceY) < distanceY);

    deltaX = computeHomeDelta(moveX, getX(), target.getX());
    deltaY = computeHomeDelta(moveY, getY(), target.getY());
}

private int computeHomeDelta(boolean move, int current, int home) {
    if (move) {
        if (current > home) return -SPEED;
        else return SPEED;
    }
    else return 0;
}

private int adjustSpeed(int speed) {
    speed = speed + Greenfoot.getRandomNumber(2 * SPEED - 1) - SPEED + 1;
    return capSpeed(speed);
}

private int capSpeed(int speed) {
    if (speed < -SPEED) return -SPEED;
    else if (speed > SPEED) return SPEED;
    else return speed;
}

private boolean randomChance(int percent) { return Greenfoot.getRandomNumber(100) < percent; }
}
*****
```

## Chapter 9 Worksheet

```
public class Ant extends Creature {

    public Ant(AntHill home) { setHomeHill(home); }

    public void act() { }

    private boolean atHome() {
        if (getHomeHill() != null) { return (Math.abs(getX() - getHomeHill().getX()) < 4) && (Math.abs(getY() - getHomeHill().getY()) < 4); }
        else { return false; }
    }
}
*****

public class Counter extends Actor {
    private int value = 0;
    private String text;

    public Counter() { this(""); }

    public Counter(String prefix) {
        text = prefix;
        int imageWidth= (text.length() + 2) * 10;
        setImage(new GreenfootImage(imageWidth, 16));
        updateImage();
    }

    public void increment() {
        value++;
        updateImage();
    }

    private void updateImage() {
        GreenfootImage image = getImage();
        image.clear();
        image.drawString(text + value, 1, 12);
    }
}
*****

public class Pheromone extends Actor {
    private final static int MAX_INTENSITY = 180;

    public Pheromone() {
        intensity = MAX_INTENSITY;
        updateImage();
    }

    public void act() { }

    private void updateImage() {
        int size = intensity / 3 + 5;
        GreenfootImage image = new GreenfootImage(size + 1, size + 1);
        int alpha = intensity / 3;
        image.setColor(new Color(255, 255, 255, alpha));
        image.fillOval(0, 0, size, size);
        image.setColor(Color.DARK_GRAY);
        image.fillRect(size / 2, size / 2, 2, 2); // small dot in the middle
        setImage(image);
    }
}
```

## Chapter 9 Worksheet

(1) What happened when the act method of the Ant class looked like each of the following?

```
public void act() {  
    randomWalk();  
}
```

```
public void act() {  
    goTowardsHome();  
}
```

```
public void act() {  
    goAwayFromHome();  
}
```

(2) So, we will have the Ant walk around and check to see if it finds some food. Write a method named checkForFood that causes the entire scenario to stop if the ant collides with a Food object.

(3) Change the code for checkForFood so that instead of stopping the scenario, the image of the ant changes to ant-with-food.gif.

```
private void checkForFood() {  
    if(getOneIntersectingObject(Food.class) != null) {  
        Greenfoot.stop();  
    }  
}
```

(4) Write the code to declare an instance variable whose type is Boolean and whose name is carryingFood.

(5) Write the code to assign the initial value false to the variable you created in question 4.

(6) Modify the code for checkForFood so that when the food is found, the variable carryingFood is set to an appropriate value.

```
private void checkForFood() {  
    if(getOneIntersectingObject(Food.class) != null) {  
        setImage("ant-with-food.gif");  
    }  
}
```



## Chapter 9 Worksheet

(10) Modify the code for the Pheromone to include an instance variable whose name is intensity. Inside of act(), intensity should decrease by one every time act() is called and when intensity is zero, the Pheromone should disappear from the world. If the intensity is not yet zero, the method updateImage should be called.

```
public class Pheromone extends Actor {
    private final static int MAX_INTENSITY = 180;

    public Pheromone() {
        intensity = MAX_INTENSITY;
        updateImage();
    }

    public void act() {

    }
}
```

(11) Note the change in the code for act() so that the method dropPheromone is called. Fill in the code for that method so that it drops a Pheromone as the Ant is walking.

```
public void act() {
    if(carryingFood) {
        walkTowardsHome();
        dropPheromone();
        checkForHome();
    }
    else {
        randomWalk();
    }
    checkForFood();
}

private void dropPheromone() {

}
}
```

## Chapter 9 Worksheet

(12) Change the code for dropPheromone so that it adds one to an instance variable every time. (You will need to create and initialize the instance variable.) If that value reaches 18, add a new Pheromone and reset the value back to zero.

```
public class Ant extends Actor {

    public Ant() {

    }

    public void act() {
        if(carryingFood) {
            walkTowardsHome();
            dropPheromone();
            checkForHome();
        }
        else {
            randomWalk();
        }
        checkForFood();
    }

    private void dropPheromone() {

    }

}
```

(13) Add more code to the Ant for another instance variable named smelledPheromone that is a boolean and is initially set to false. Adjust act so that if the ant smells a pheromone it walks away from home. Otherwise, it looks to see if it collided with a pheromone, and if so, sets smelledPheromone to true.

```
public class Ant extends Actor {

    public Ant() {

    }

    public void act() {
        if(carryingFood) {
            walkTowardsHome();
            dropPheromone();
            checkForHome();
        }
        else {
            randomWalk();
        }
        checkForFood();
    }

}
```