Snake game
Inspired by zetcode.com Java 2D games

Snake is a video game that originated during the late 1970s in arcades becoming something of a classic. It became the standard pre-loaded game on Nokia phones in 1998.

The player controls a long, thin creature, resembling a snake, which roams around on a bordered plane, picking up food (or some other item), trying to avoid hitting its own tail or the edges of the playing area. Each time the snake eats a piece of food, its tail grows longer, making the game increasingly difficult. The user controls the direction of the snake's head (up, down, left, or right), and the snake's body follows.

Here is a quick overview of how it is designed. In the snippets below, DOT_SIZE is the width (and height) of the section of the snake say 20 pixels. (You can think of this as something similar to a square on a game board.) MAX_DOTS constant defines the maximum number of possible dots on the board. The size 600x400 defines the size of the board in pixels, so dividing that number by DOT_SIZE*DOT_SIZE computes the number of DOTS (i.e. squares) on the board. Going forward, a part of the snake will occupy one of these dots.

private final MAX_DOTS = (600*400)/(DOT_SIZE*DOT_SIZE);

Assuming the world was defined as: super(600, 400, 1)

Two arrays store the x, y coordinates of all joints of a snake.

private int x[ ] = new int [MAX_DOTS];
private int y[ ] = new int [MAX_DOTS];

If the snake eats, it grows by one more section.

The variable dots, in SnakeWorld, keeps track of how many sections we have at any time. In the above example before eating, dots = 4. After eating dots is increased to 5. Remember, in computer science, we start counting from 0 not 1.
Here we have parts = 7 indicating that we have 7 parts including the head. You control the head of the snake by using the cursor keys. Clicking the left arrow key changes the direction of the head to point East, Left to West, Up to North and Down to South. The head keeps moving in the same direction until an arrow key is clicked to change its direction.

The rest of the joints move one position up the chain. The first joint moves where the head was, the second joint moves where the first was, the third joint where the second was, etc.

```
for (int k = parts; z > 0; z--) {
    x[z] = x[z - 1];
    y[z] = y[z - 1];
}
```

The above code, in SnakeWorld, moves the joints up the chain.

```
if (Greenfoot.isKeyDown("left"))
{
    setRotation( .. );
}
```

The above code snippet shows how to change the orientation of the head when the left arrow key is pressed. Check the API for setRotation(..) to figure out what value to specify as parameter.

**Your task**

1- Download and unzip the Snake scenario. Start greenfoot and load this scenario.

2- Edit the SnakeWorld class to populate the snake world with one head and three sections, a total of four parts (dots=4). Add a piece of food (what snakes like to eat).

3- In the Dot class, make the head move and change direction when arrow keys are pressed.

4- Make the rest of the snake body follow the head.
5- When food is eaten, make the tail grow by one section and make more food (randomly 1 to 3 more) appear.

6- Go back to SnakeWorld and change things so that the when food is initially added, it is added at a random location.

7- Display a message “Game Over” if the snake hits one of its joints or touches the edges. (Hint see GreenfootImage)

8- More advanced: Add another object that snakes don’t like to eat. Make the tail shrink by one section when the snake touches it. Make another one appear.

9- Add your own modifications. Enjoy playing the game.