public class Snake {

// Stores the joints / body part locations for our snake

private final int[] x = new int[Board.getAllDots()];

private final int[] y = new int[Board.getAllDots()];

// Stores direction of our snake

private boolean movingLeft = false;

private boolean movingRight = false;

private boolean movingUp = false;

private boolean movingDown = false;

private int joints = 0; // Stores # of dots / joints the snake has (starts

// with 3)

public int getSnakeX(int index) {

return x[index];

}

public int getSnakeY(int index) {

return y[index];

}

public void setSnakeX(int i) {

x[0] = i;

}

public void setSnakeY(int i) {

y[0] = i;

}

public boolean isMovingLeft() {

return movingLeft;

}

public void setMovingLeft(boolean movingLeft) {

this.movingLeft = movingLeft;

}

public boolean isMovingRight() {

return movingRight;

}

public void setMovingRight(boolean movingRight) {

this.movingRight = movingRight;

}

public boolean isMovingUp() {

return movingUp;

}

public void setMovingUp(boolean movingUp) {

this.movingUp = movingUp;

}

public boolean isMovingDown() {

return movingDown;

}

public void setMovingDown(boolean movingDown) {

this.movingDown = movingDown;

}

public int getJoints() {

return joints;

}

public void setJoints(int j) {

joints = j;

}

public void move() {

for (int i = joints; i > 0; i--) {

// Moves the joints of the snake 'up the chain'

// Meaning, the joint of the snake all move up one

x[i] = x[(i - 1)];

y[i] = y[(i - 1)];

}

// Moves snake to the left

if (movingLeft) {

x[0] -= Board.getDotSize();

}

// To the right

if (movingRight) {

x[0] += Board.getDotSize();

}

// Down

if (movingDown) {

y[0] += Board.getDotSize();

}

// And finally up

if (movingUp) {

y[0] -= Board.getDotSize();

}

// Dotsize represents the size of the joint, so a pixel of DOTSIZE

// gets added on to the snake in that direction

}

}