public class MovingNode extends Node implements ClockListener{

public MovingNode(){

Clock.addClockListener(this, 2);

setDirection(Math.random()\*2\*Math.PI);

}

public void onClock(){

move(2);

}

}

**Clock and Timing:** As you can see, the new class implements the [ClockListener](http://jbotsim.sourceforge.net/javadoc/index.html?jbotsim/event/ClockListener.html" \t "doc) interface, which consists of a single method onClock()that JBotSim will call periodically. Hence, registering with Clock.addClockListener(this,2) at construction time implies this node will subsequently have its onClock() method called every 2 units of time (the default unit is 10ms).

public class LonerMessageBased extends Node implements MessageListener{

public LonerMessageBased(){

addMessageListener(this);

setColor("green");

}

public void onMessage(Message msg) {

isAlone = false;

}

}

**Message:** As per the reception, there is essentially two ways to proceed. The node might ask to be notified in an event-based fashion -- which is the case here -- or it may scrutinize its mailbox manually. In the first case, the node first registers as its own message listener by calling this.addMessageListener(this) in the constructor. Then every time a message is received, it is notified through the onMessage() method, where msg is an object of type [Message](http://jbotsim.sourceforge.net/javadoc/index.html?jbotsim/Message.html) that contains, in itscontent field, the reference to the sent object. Here we simply ignore the content of the message, as the mere reception of it implies there exists some neighbors.