

```

1 package info.openrocket.core.simulation.listeners;
2
3 import info.openrocket.core.aerodynamics.AerodynamicForces;
4 import info.openrocket.core.aerodynamics.FlightConditions;
5 import info.openrocket.core.masscalc.RigidBody;
6 import info.openrocket.core.models.atmosphere.AtmosphericConditions;
7 import info.openrocket.core.motor.MotorConfigurationId;
8 import info.openrocket.core.rocketcomponent.MotorMount;
9 import info.openrocket.core.rocketcomponent.RecoveryDevice;
10 import info.openrocket.core.simulation.AccelerationData;
11 import info.openrocket.core.simulation.FlightEvent;
12 import info.openrocket.core.simulation.MotorClusterState;
13 import info.openrocket.core.simulation.SimulationStatus;
14 import info.openrocket.core.simulation.exception.SimulationException;
15 import info.openrocket.core.util.BugException;
16 import info.openrocket.core.util.Coordinate;
17
18 /**
19  * An abstract base class for implementing simulation listeners. This class
20  * implements all
21  * of the simulation listener interfaces using methods that have no effect on
22  * the simulation.
23  * The recommended way of implementing simulation listeners is to extend this
24  * class.
25  *
26  * @author Sampo Niskanen <sampo.niskanen@iki.fi>
27  */
28 public class AbstractSimulationListener implements SimulationListener, SimulationComputationListener,
29     SimulationEventListener, Cloneable {
30
31     /// SimulationListener ///
32
33     @Override
34     public void startSimulation(SimulationStatus status) throws SimulationException {
35         // No-op
36     }
37
38     @Override
39     public void endSimulation(SimulationStatus status, SimulationException exception) {
40         // No-op
41     }
42
43     @Override
44     public boolean preStep(SimulationStatus status) throws SimulationException {
45         return true;
46     }
47
48     @Override
49     public void postStep(SimulationStatus status) throws SimulationException {
50         // No-op
51     }
52
53     /**
54      * {@inheritDoc}
55      * <p>
56      * <em>This implementation of the method always returns <code>false</code>.</em>
57      */
58     @Override
59     public boolean isSystemListener() {
60         return false;
61     }
62
63     /// SimulationEventListener ///
64
65     @Override
66     public boolean addFlightEvent(SimulationStatus status, FlightEvent event) throws SimulationException {
67         return true;
68     }
69
70     @Override
71     public boolean handleFlightEvent(SimulationStatus status, FlightEvent event) throws SimulationException {
72         return true;
73     }
74
75     @Override
76     public boolean motorIgnition(SimulationStatus status, MotorConfigurationId motorId, MotorMount mount,
77         MotorClusterState instance) throws SimulationException {
78         return true;
79     }
80
81     @Override
82     public boolean recoveryDeviceDeployment(SimulationStatus status, RecoveryDevice recoveryDevice)
83         throws SimulationException {
84         return true;
85     }
86
87     /// SimulationComputationListener ///
88
89     @Override
90     public AccelerationData preAccelerationCalculation(SimulationStatus status) throws SimulationException {
91         return null;

```

```

92     }
93
94     @Override
95     public AerodynamicForces preAerodynamicCalculation(SimulationStatus status) throws SimulationException {
96         return null;
97     }
98
99     @Override
100    public AtmosphericConditions preAtmosphericModel(SimulationStatus status) throws SimulationException {
101        return null;
102    }
103
104    @Override
105    public FlightConditions preFlightConditions(SimulationStatus status) throws SimulationException {
106        return null;
107    }
108
109    @Override
110    public double preGravityModel(SimulationStatus status) throws SimulationException {
111        return Double.NaN;
112    }
113
114    @Override
115    public RigidBody preMassCalculation(SimulationStatus status) throws SimulationException {
116        return null;
117    }
118
119    @Override
120    public double preSimpleThrustCalculation(SimulationStatus status) throws SimulationException {
121        return Double.NaN;
122    }
123
124    @Override
125    public Coordinate preWindModel(SimulationStatus status) throws SimulationException {
126        return null;
127    }
128
129    @Override
130    public AccelerationData postAccelerationCalculation(SimulationStatus status, AccelerationData acceleration)
131        throws SimulationException {
132        return null;
133    }
134
135    @Override
136    public AerodynamicForces postAerodynamicCalculation(SimulationStatus status, AerodynamicForces forces)
137        throws SimulationException {
138        return null;
139    }
140
141    @Override
142    public AtmosphericConditions postAtmosphericModel(SimulationStatus status,
143        AtmosphericConditions atmosphericConditions) throws SimulationException {
144        return null;
145    }
146
147    @Override
148    public FlightConditions postFlightConditions(SimulationStatus status, FlightConditions flightConditions)
149        throws SimulationException {
150        return null;
151    }
152
153    @Override
154    public double postGravityModel(SimulationStatus status, double gravity) throws SimulationException {
155        return Double.NaN;
156    }
157
158    @Override
159    public RigidBody postMassCalculation(SimulationStatus status, RigidBody rigidBody) throws SimulationException {
160        return null;
161    }
162
163    @Override
164    public double postSimpleThrustCalculation(SimulationStatus status, double thrust) throws SimulationException {
165        return Double.NaN;
166    }
167
168    @Override
169    public Coordinate postWindModel(SimulationStatus status, Coordinate wind) throws SimulationException {
170        return null;
171    }
172
173    @Override
174    public AbstractSimulationListener clone() {
175        try {
176            return (AbstractSimulationListener) super.clone();
177        } catch (CloneNotSupportedException e) {
178            throw new BugException(e);
179        }
180    }
181
182 }

```