

```
1  /*****
2  *   On trouve dans cette onglet des variables globales
3  *   et des procédures pour la gestion du projet.
4  *
5  *****/
6
7  boolean terminate=false;  8
9  final int samplingTime=10;

10
11
12 // Liste des acteurs, accès via un numéro de 0 à n-1 (si on à n acteurs)
13 ArrayList<Actor> actorsContainer = new ArrayList<Actor>();
14
15 // liste des noms d'acteurs et de l'index de ceux-ci dans actorsContainer
16 HashMap<String, Integer> actorsIndex = new HashMap<String, Integer>(); // permettra
d'agir sur un acteur via son nom.
17
18 // liset des acteurs sensible au click de souris, il doit implémenter l'interface
Clickable.
19 ArrayList<Clickable> mouseSensitivActors = new ArrayList<Clickable>();
20 int lastMouseSensitivActors=-1;
21 // index de l'acteur qui vient de réagir à mousePressed, il est le seul qui peut
réagir à mouseRealease
```

```
22
23
24 ArrayList<Runnable> runnableActors = new ArrayList<Runnable>();
25
26 TerminalBoard  bornier;
27
28 //*****
29 // procédure de chargement des acteurs, reçoit en paramètre un tableau d'objet au
format JSON
30 void loadActors(JSONArray actors) {
31     Actor actor;
32     for (int i=0; i<actors.size(); i++) {
33         actor=null;
34         JSONObject jActor=actors.getJSONObject(i);
35         String name=jActor.getString("name");
36         String type=jActor.getString("type");
37
38         actorsIndex.put(name, i);
39
40         // pour les acteurs de base
41         if (type.equals("Actor")) {
42             actor=new Actor(jActor);
43             actorsContainer.add(actor);
44         }
```

```
45
46     // pour les Buttons
47     if (type.equals("Button")) {
48         actor=new Button(jActor);
49         actorsContainer.add(actor);
50         mouseSensitivActors.add((Button)actor);
51     }
52
53     // pour les Indicators
54     if (type.equals("Indicator")) {
55         actor=new Indicator(jActor);
56         actorsContainer.add(actor);
57     }
58
59     //HorizontalPositionSensor
60     if (type.equals("HorizontalPositionSensor")) {
61         actor=new HorizontalPositionSensor(jActor);
62         actorsContainer.add(actor);
63         runnableActors.add((HorizontalPositionSensor)actor);
64     }
65
66     //VerticalPositionSensor
67     if (type.equals("VerticalPositionSensor")) {
68         actor=new VerticalPositionSensor(jActor);
```

```
69     actorsContainer.add(actor);
70     runnableActors.add((VerticalPositionSensor)actor);
71 }
72
73 //VerticalColorSensor
74 if (type.equals("VerticalColorSensor")) {
75     actor=new VerticalColorSensor(jActor);
76     actorsContainer.add(actor);
77     runnableActors.add((VerticalColorSensor)actor);
78 }
79
80 // HorizontalMovingActor Wagon
81 if (type.equals("HorizontalMovingActor")) {
82     actor=new HorizontalMovingActor(jActor);
83     actorsContainer.add(actor);
84     runnableActors.add((HorizontalMovingActor)actor);
85 }
86
87 // VerticalMovingActor Hook
88 if (type.equals("VerticalMovingActor")) {
89     actor=new VerticalMovingActor(jActor);
90     actorsContainer.add(actor);
91     runnableActors.add((VerticalMovingActor)actor);
92 }
```

```
93
94     // Follower cleanBox, dirtyBox
95     if (type.equals("Follower")) {
96         actor=new Follower(jActor);
97         actorsContainer.add(actor);
98     }
99
100    // Tank
101    if (type.equals("Tank")) {
102        actor=new Tank(jActor);
103        actorsContainer.add(actor);
104        runnableActors.add((Tank)actor);
105    }
106
107    // Valve
108    if (type.equals("Valve")) {
109        actor=new Valve(jActor);
110        actorsContainer.add(actor);
111        runnableActors.add((Valve)actor);
112    }
113
114    // Ball
115    if (type.equals("Ball")) {
116        actor=new Ball(jActor);
```

C:/nqCbtp/JavaApplication8/src/Terminal.java

```
117         actorsContainer.add(actor);
118         runnableActors.add((Ball)actor);
119     }
120
121     //HorizontalPneumaticCylinder
122     if (type.equals("HorizontalPneumaticCylinder")) {
123         actor=new HorizontalPneumaticCylinder(jActor);
124         actorsContainer.add(actor);
125         runnableActors.add((HorizontalPneumaticCylinder)actor);
126     }
127
128     // LevelPositionSensor
129     if (type.equals("LevelPositionSensor")) {
130         actor=new LevelPositionSensor(jActor);
131         actorsContainer.add(actor);
132         runnableActors.add((LevelPositionSensor)actor);
133     }
134
135     //PLC
136     if (type.equals("Plc")) {
137         actor=new Plc(jActor);
138         actorsContainer.add(actor);
139         runnableActors.add((Plc)actor);
140     }
```

```
141
142
143     if (actor==null) {
144         print("Erreur, l'acteur ");
145         print(name);
146         print(" dont le type déclaré est ");
147         print(type);
148         println(" ,ce type est inconnu.");
149         exit();
150         return;
151     }
152 }
153 }
154
155 void loadTerminalBoard(JSONArray terminals)
{ 156     bornier=new TerminalBoard(terminals);
157     runnableActors.add(bornier);
158 }
159
160
161 void execute(String action) {
162     if (action.equals("quit"))
163         terminate=true;
164 }
```