

Cas d'étude du robot (version 1)

Jérôme Delatour

Expérimenter le cas d'étude

Lancement du simulateur « intox »

```
jerome@T420:~$ java -jar intox-java_1.8-v0.3.6.jar &
```

Cliquer sur démarrer pour activer le simulateur, il est alors en attente d'ordre de déplacement.

Compilation de l'exemple « hello_robot-v0.8 »,

On modifie si besoin le fichier Makefile du répertoire principal.

Afin de positionner la variable INTOXDIR sur la librairie infox (infox permet de dialoguer avec intox).

Par défaut, la librairie est considérée comme à coté du dossier exemple « hello_robot-v0.8 »

Attention : assurez vous de ne pas avoir d'espace ou de caractères accentués dans les noms de vos répertoires et vos chemins.

On utilise l'utilitaire make pour faire la compilation

```
jerome@T420: make
```

On rend exécutable la première fois le fichier robot_pc ainsi généré, puis on le lance

```
jerome@T420: ./bin/robot_pc
```

Si tout va bien, le robot se met à tourner tout seul

Démonstration sur l'exemple

Démonstration

A faire avant le début du S8 (option SE)

Robot V1

Vous pouvez faire les actions suivantes :

q:aller à gauche

d:aller à droite

z:avancer

s:reculer

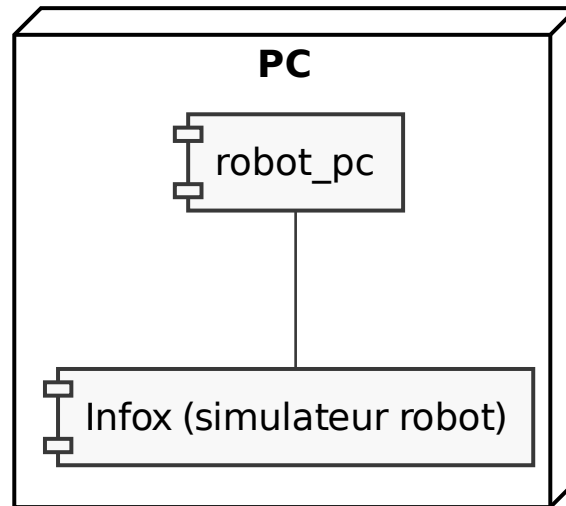
:stopper

e:effacer les log

r:afficher l'état du robot

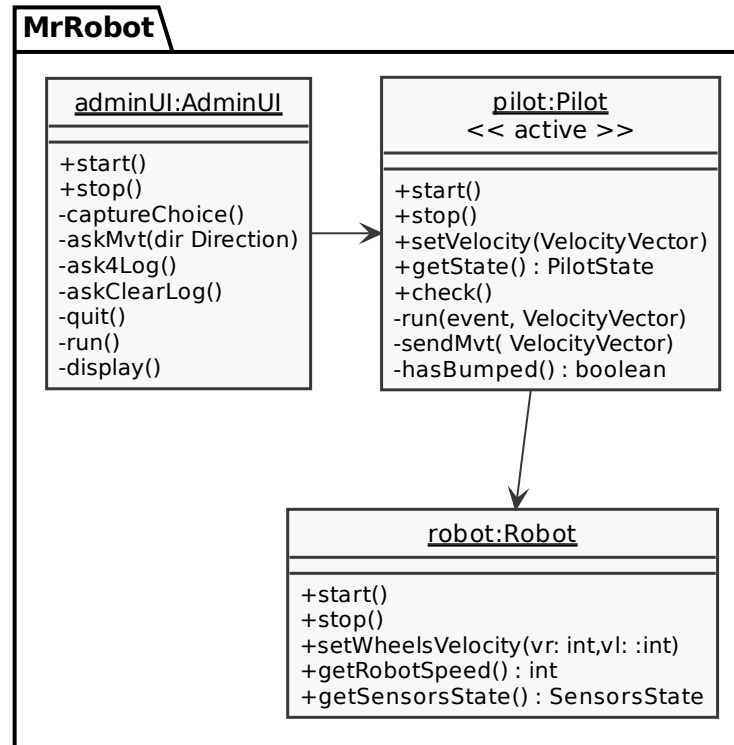
a:quitter

Architecture matérielle de mrRobot, version 1

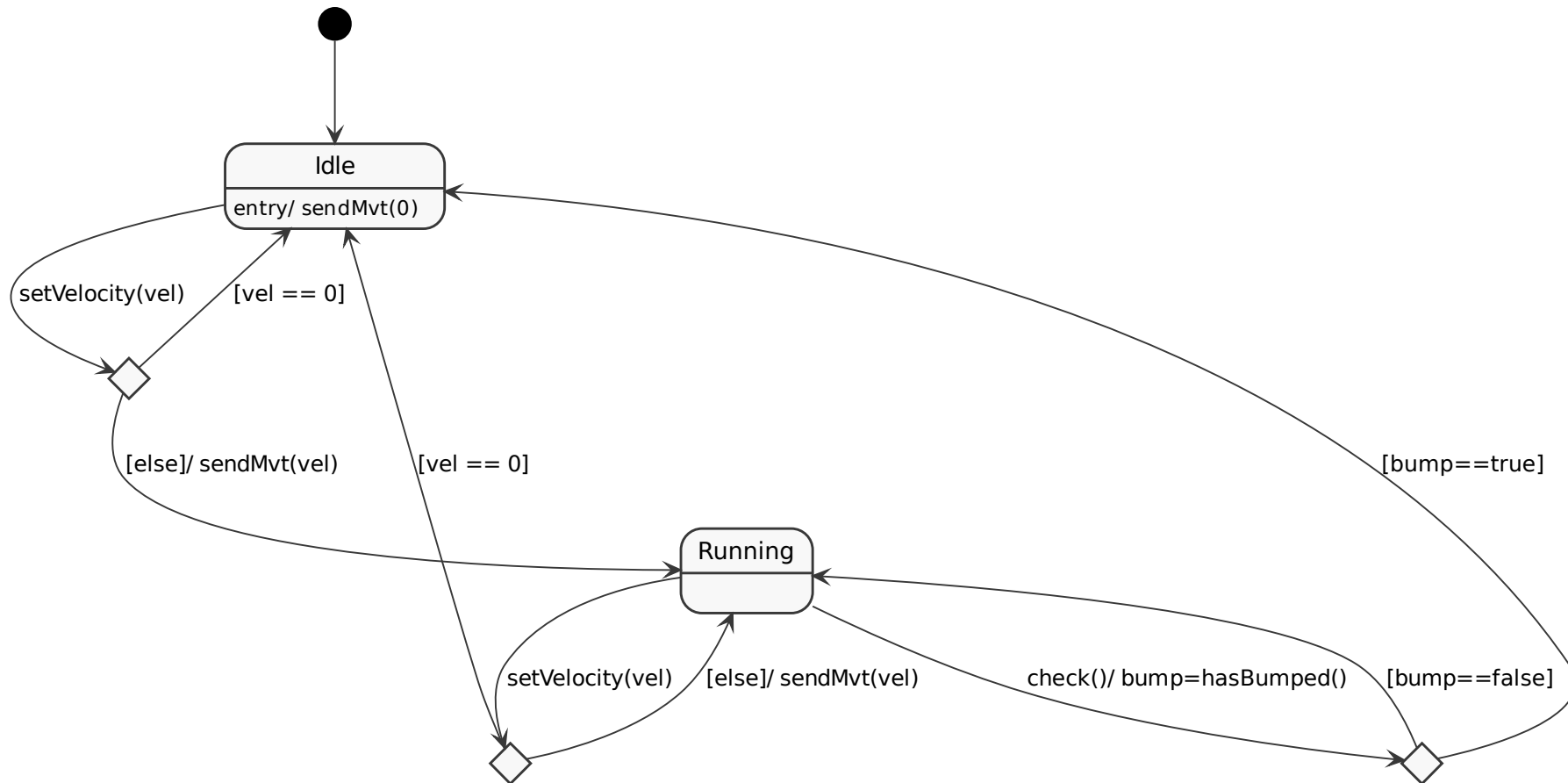


Conception générale

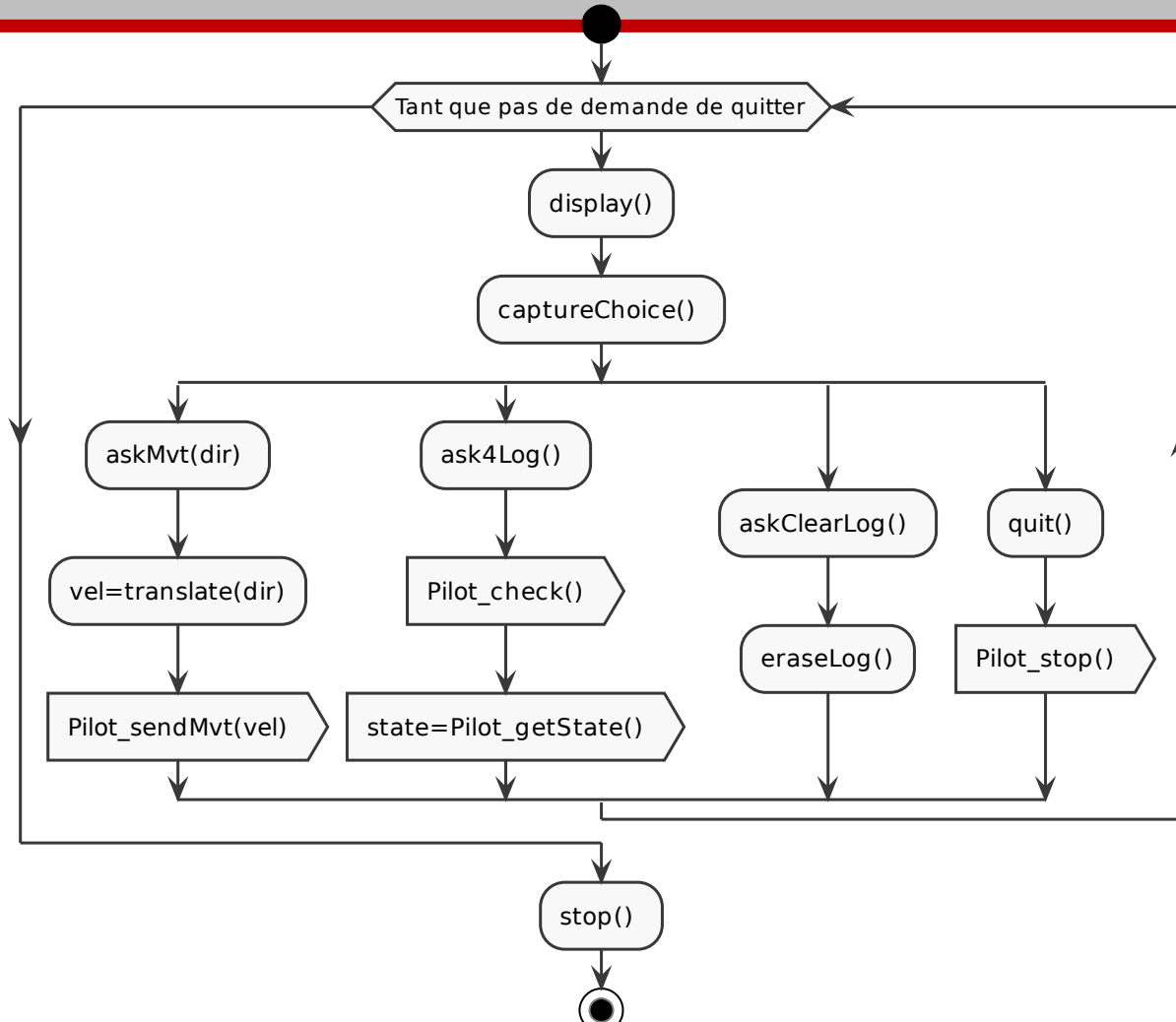
Architecture logique V1 de MrRobot, représenté par un diagramme non UML "pure"
(un mélange de diagramme à objet et classe, utilisation toutefois souvent tolérée)



Comportement de Pilote V1,
représenté par un diagramme état-transition



Méthode run de AdminUI




```
int main (int argc, char *argv[])  
{  
  
    AdminUI_new();  
    AdminUI_start();  
    AdminUI_stop();  
    AdminUI_free();  
    return 0;  
}
```