



30/03/2021

Dossier de conception

Équipe Élan

Objectif mars – UML

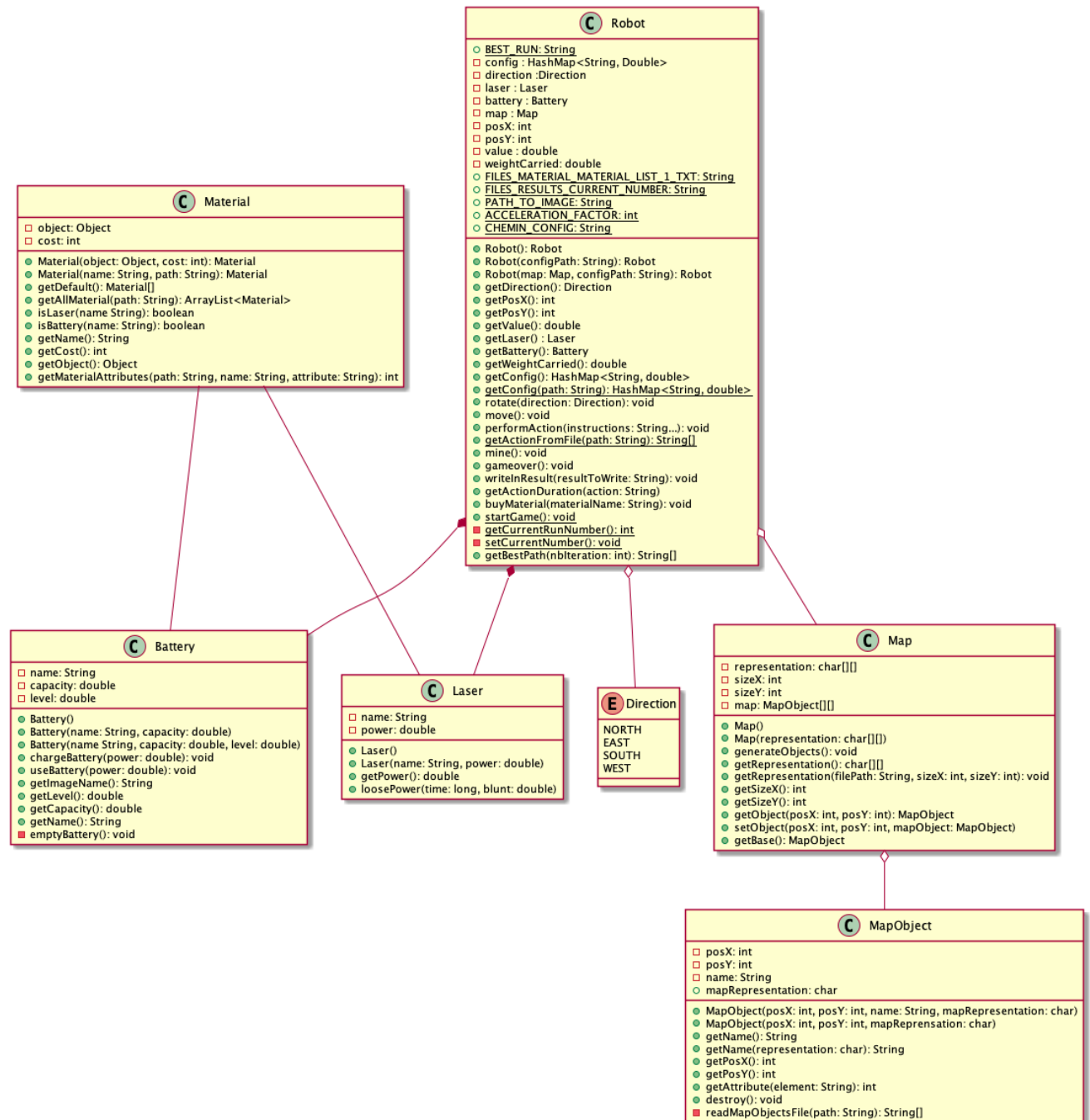
Table des matières

Partie 1	2
Digramme classe	2
Digramme cas d'utilisation	3
Digramme état-transition bouger robot	4
Diagramme état-transition acheter matériel	5
Diagramme séquence avancer robot	7
Partie 2	9
Diagramme classe	9
Diagramme cas d'utilisation	10
Diagramme état transition instruction Y	11
Diagramme séquence	12

Partie 1

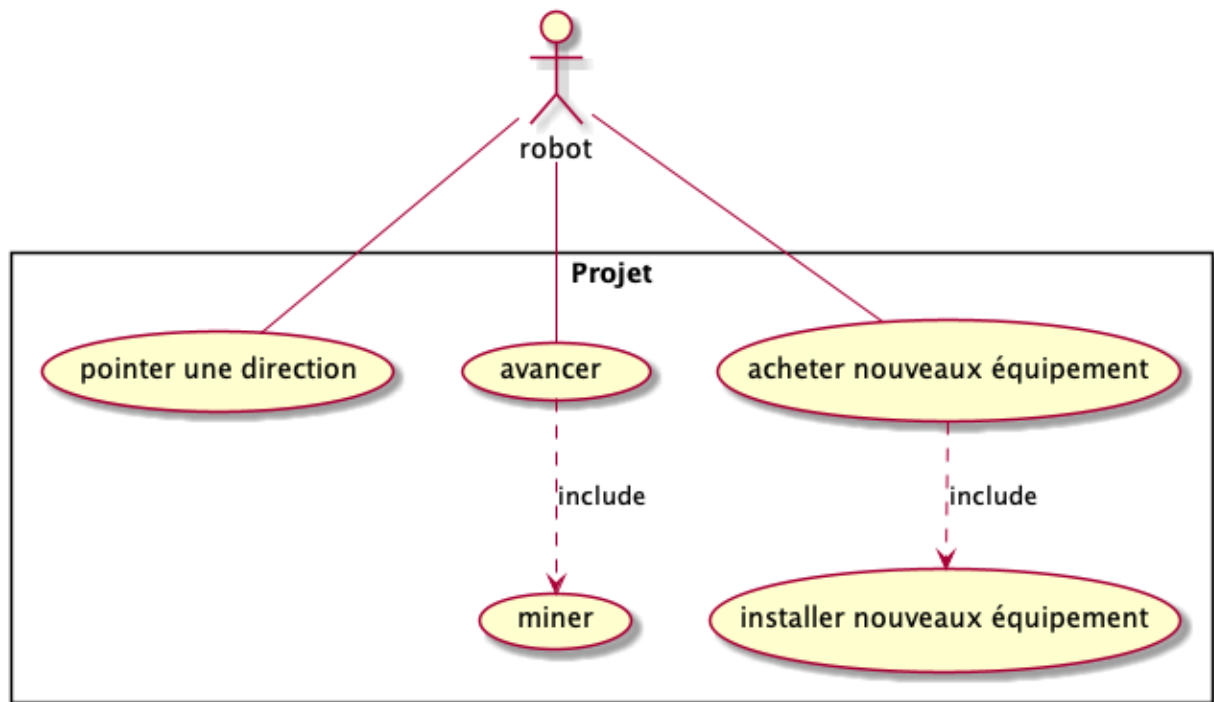
Digramme classe

digramme classe partie 1



Le digramme classe a été représenter de cette façon pour montrer les classes qui gèrent le robot, j'ai pas représenter les classes qui gèrent l'interface graphique.

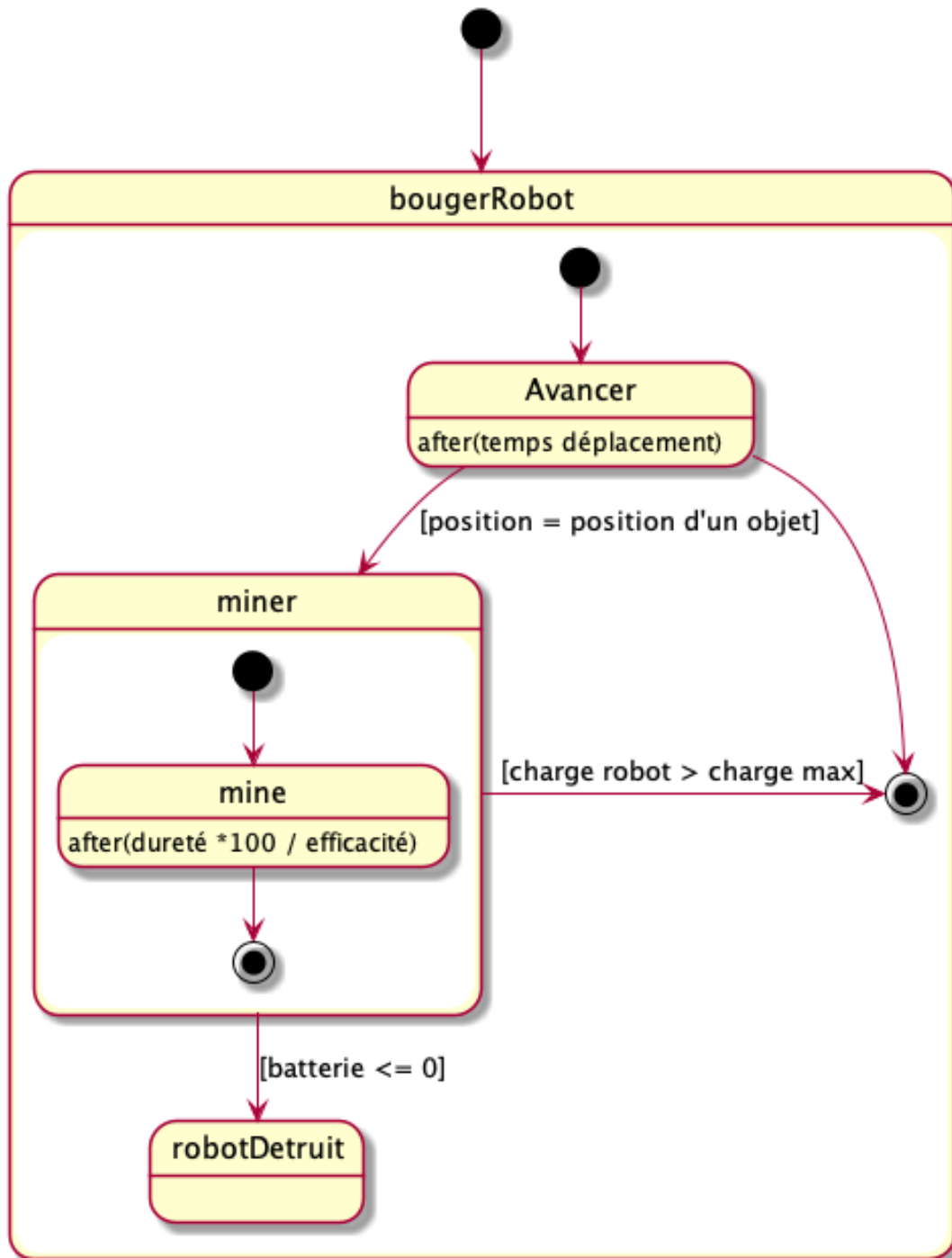
Diagramme cas partie 1



Pour le diagramme Cas d'utilisation, j'ai pris en compte ce que fait notre code, le robot peut directement :

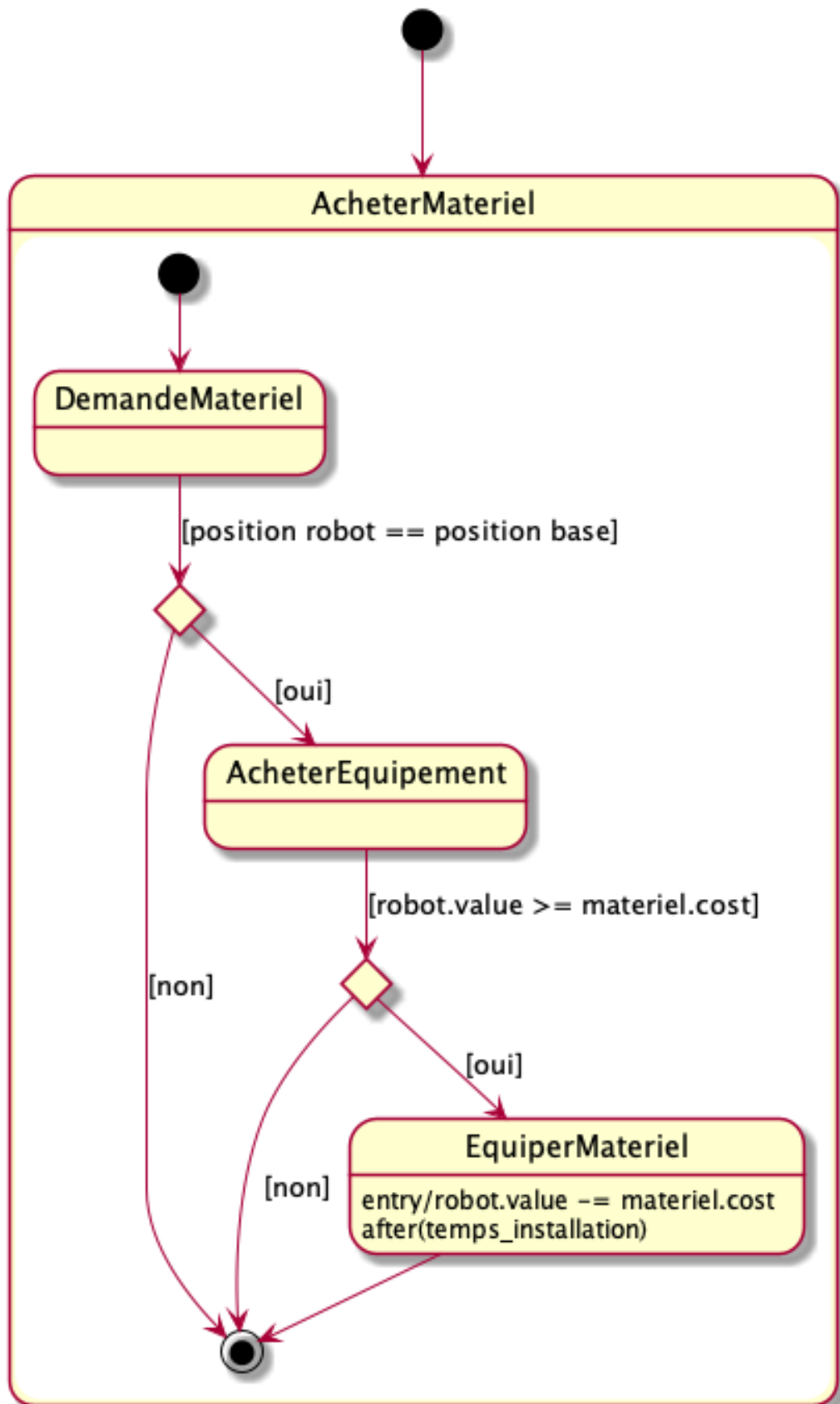
- Pointer une direction
- Avancer. Lorsque le robot avance, il mine la case sur laquelle il s'est avancé si elle n'est pas vide
- Acheter un nouvel équipement. Il faut que le robot soit sur la base pour effectuer cette action. Et lorsqu'il achète l'équipement, il l'installe automatiquement. (Le robot ne peut pas acheter un équipement sans l'installer.

diagramme etat transition bouger robot



Pour le digramme d'état transition lorsque le robot veut avancer. Il avance avec le temps de déplacement à vide puis finit. Si le robot est sur un objet, il mine (qui remplace alors le temps de déplacement) avec comme durée, cent fois la dureté de l'objet divisé par l'efficacité du laser. Si la batterie devient vide ou négative, le robot est détruit. Si la charge max est atteinte, le robot s'arrête.

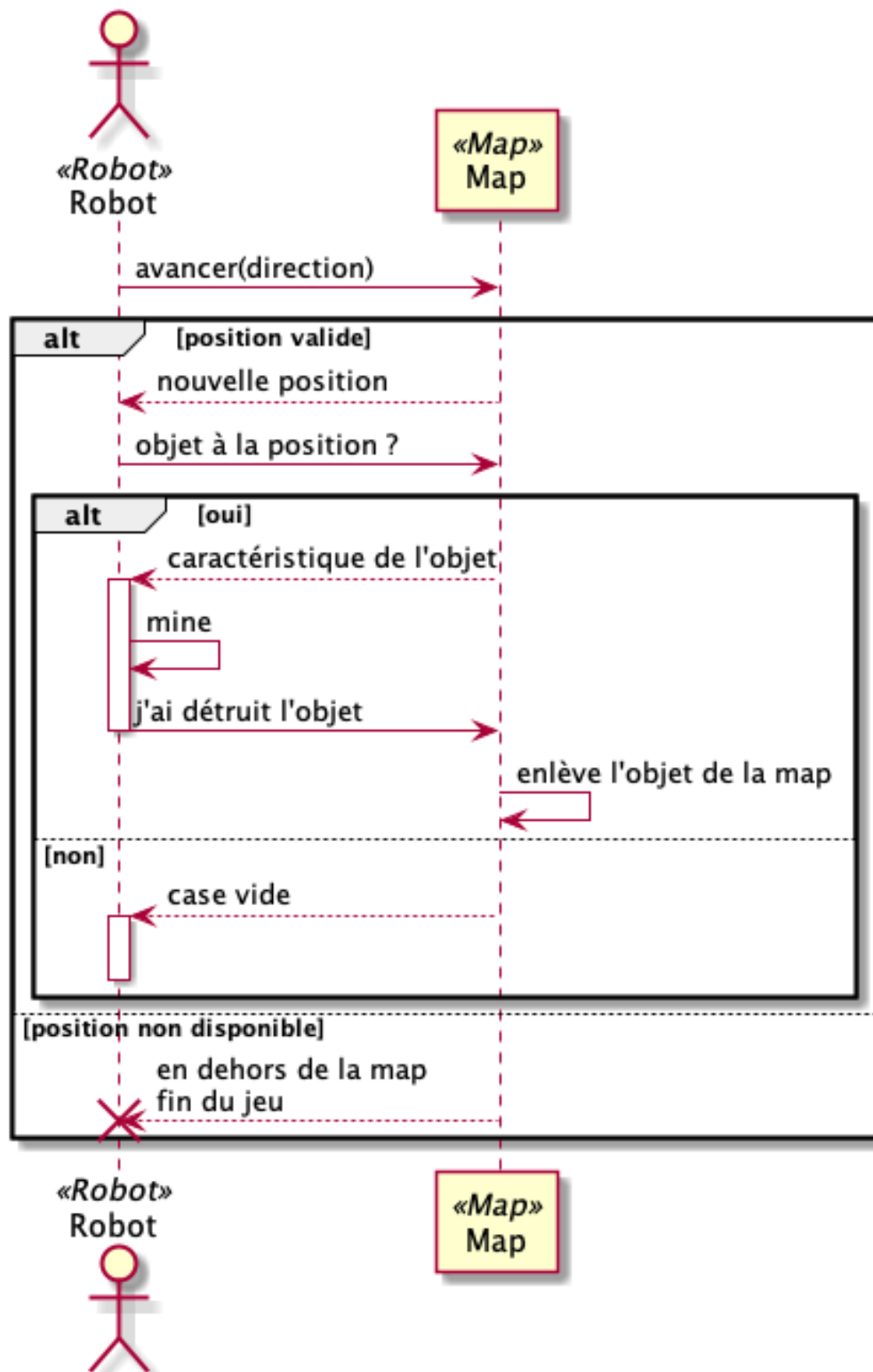
diagramme etat transition acheter matériel



Pour le diagramme état transition acheter matériel, on teste si le robot est bien sur la base, s'il ne l'est pas, rien ne se passe. L'action se termine. S'il est bien sur la base, on regarde quel équipement il veut acheter, s'il a les ressources nécessaires pour l'acheter on l'équipe avec. Ça prend temps_installation de temps d'installer le matériel. Sinon, il ne se passe rien et l'action est fini.

De plus, lorsque le robot se fait équiper, on lui enlève les ressources correspondant au prix du nouveau matériel acquis.

diagramme sequence partie 1



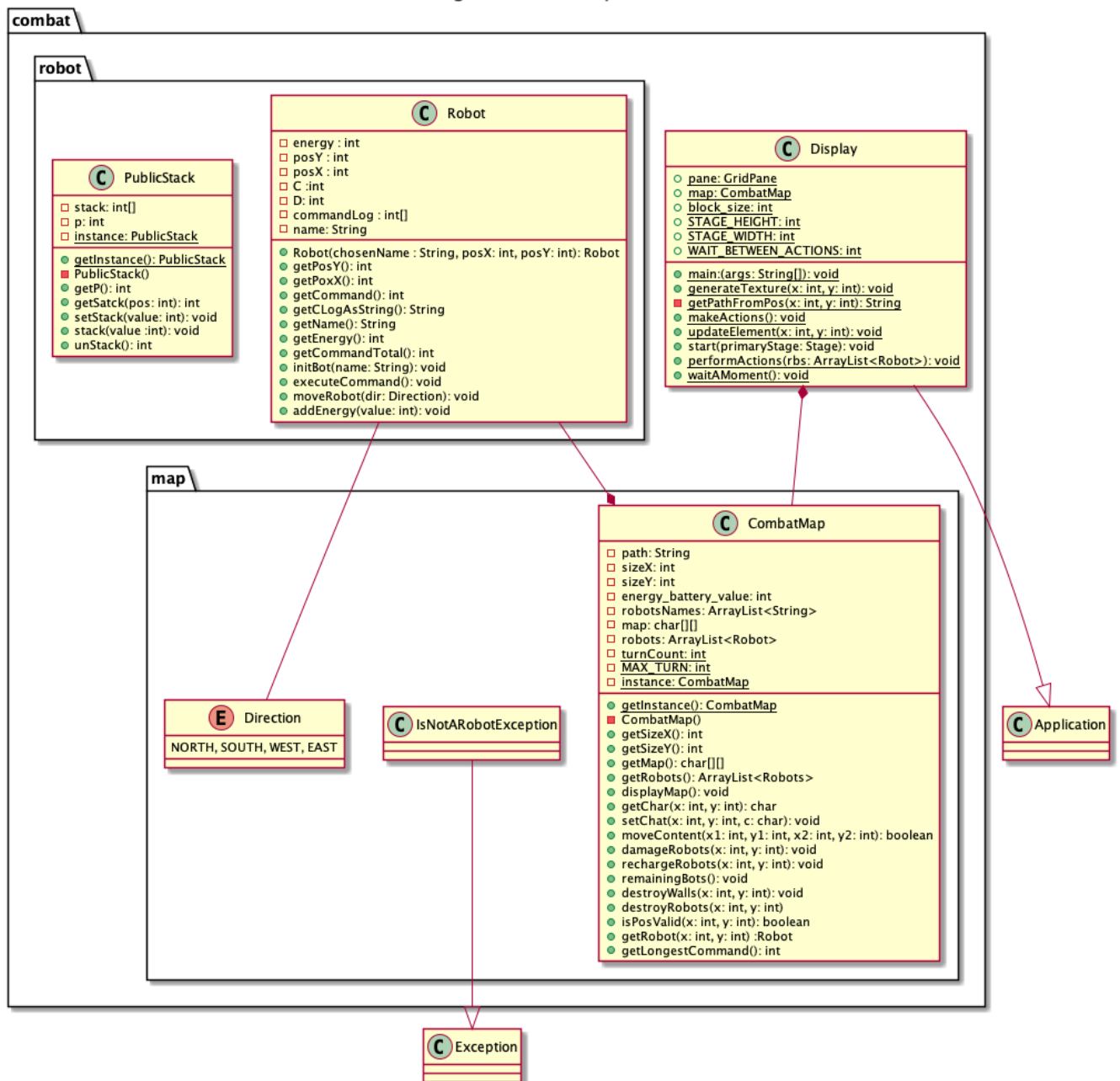
Pour le diagramme séquence de la partie 1, j'ai décidé de modéliser la séquence lorsque le robot veut avancer. La durée lorsqu'il arrive sur un objet correspond à la durée de minage, la durée lorsqu'il arrive sur une case vide correspond à la durée d'un déplacement normal. S'il

veut avancer sur une case non disponible, en dehors de la carte par exemple, le robot est détruit.

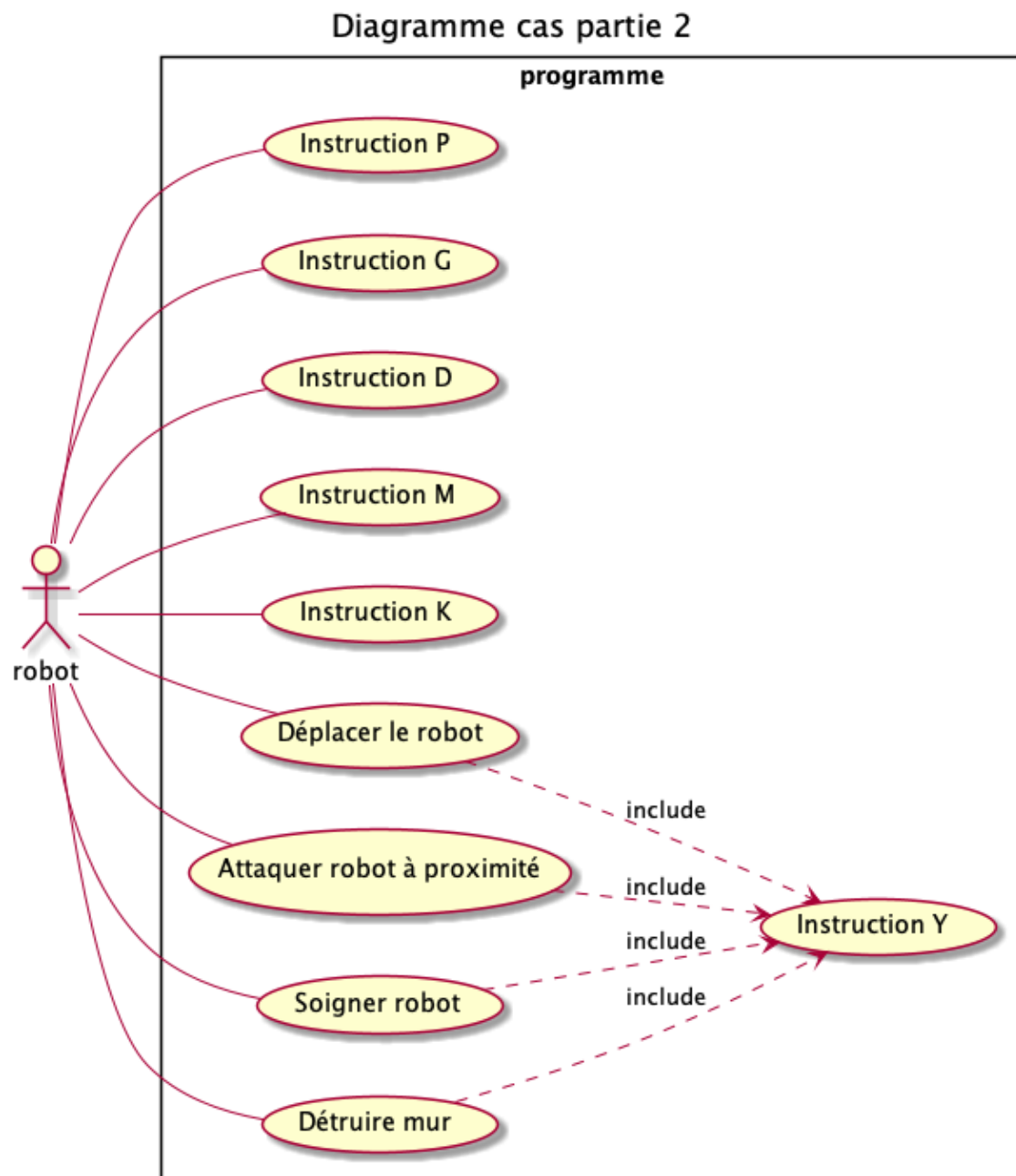
Partie 2

Diagramme classe

Diagramme classe partie 2

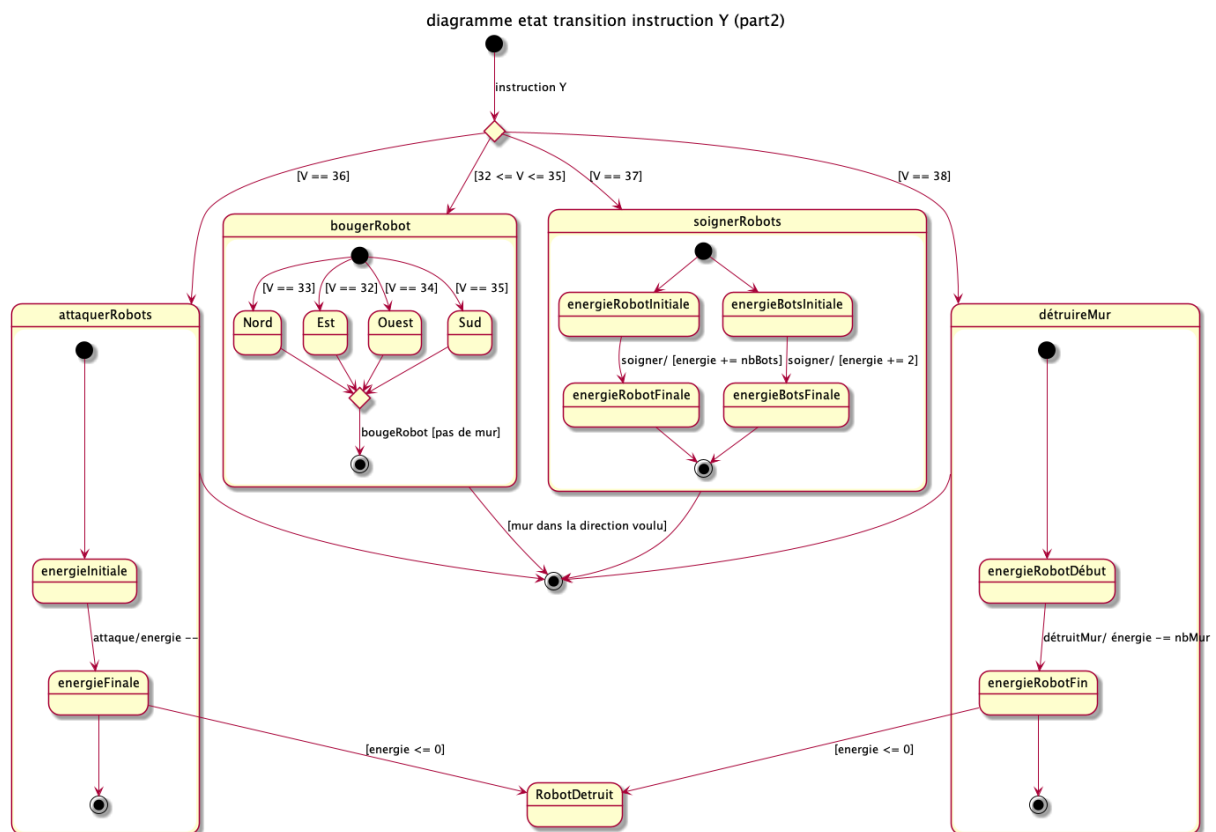


Le diagramme classe de la partie 2 représente toutes les classes de la partie. Incluant les classes qui gèrent les combats entre robots, les robots et l'interface graphique.



Pour le diagramme séquence de la partie 2, le robot peut faire toutes les actions présente dans le sujet. L'action déplacer est la concaténation des avancer haut, bas, droite, gauche. Les actions Déplacer, attaquer, soigner et détruire sont appelées avec l'Instruction Y, que j'ai représenté avec un « include ». J'ai décidé de rester très général dans ce diagramme et de ne pas détailler chaque cas d'utilisation.

Diagramme état transition instruction Y



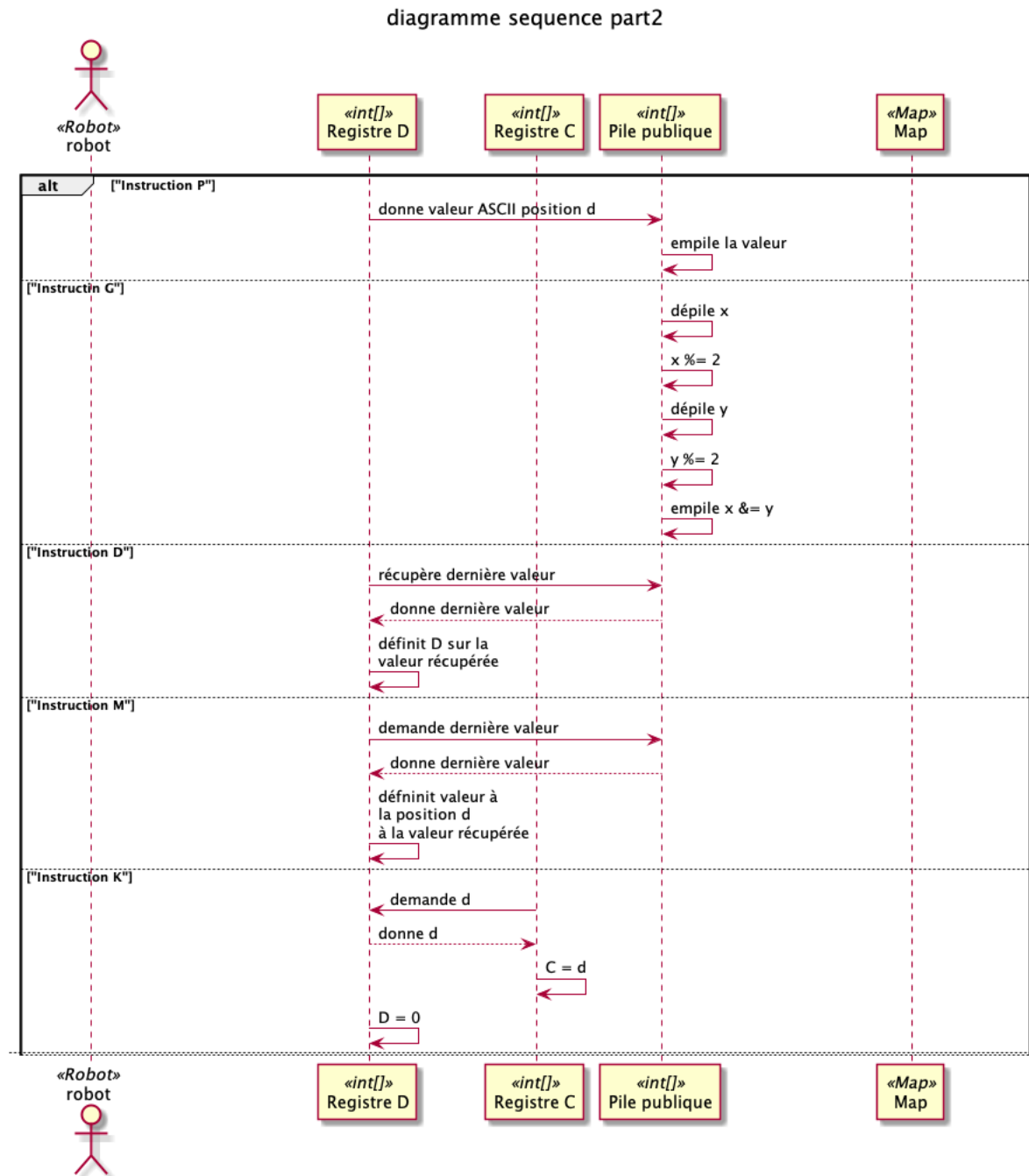
Pour le digramme d'état transition de la partie 2, j'ai détaillé l'instruction Y du robot (il n'y a qu'un seul digramme, vous m'avez dit qu'il était suffisant pour ne pas en faire un deuxième) Lorsque le robot lance l'instruction Y, en fonction de la valeur de V, il va effectuer différentes actions.

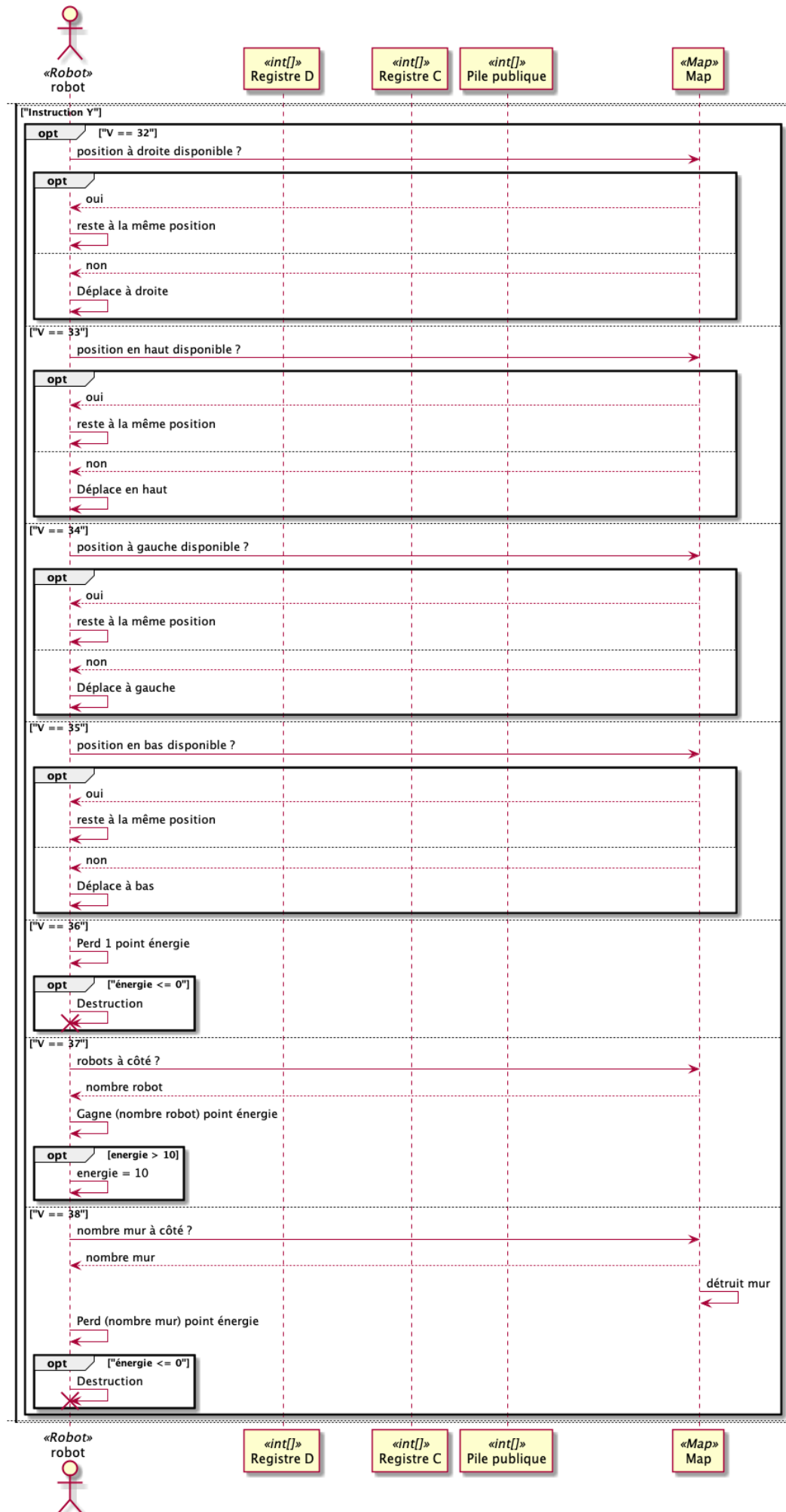
Pour V entre 32 et 35, le robot va se déplacer, en fonction de la valeur de V, en haut, en bas, à droite ou à gauche. S'il n'y a pas de mur dans la direction voulu il avance, sinon il ne fait rien. Si V = 36, le robot va attaquer les robots à côté de lui, il perd alors 1 point d'énergie. Si l'énergie est toujours supérieure à 0 il finit correctement, si, à l'inverse, l'énergie est égale à 0, le robot est détruit.

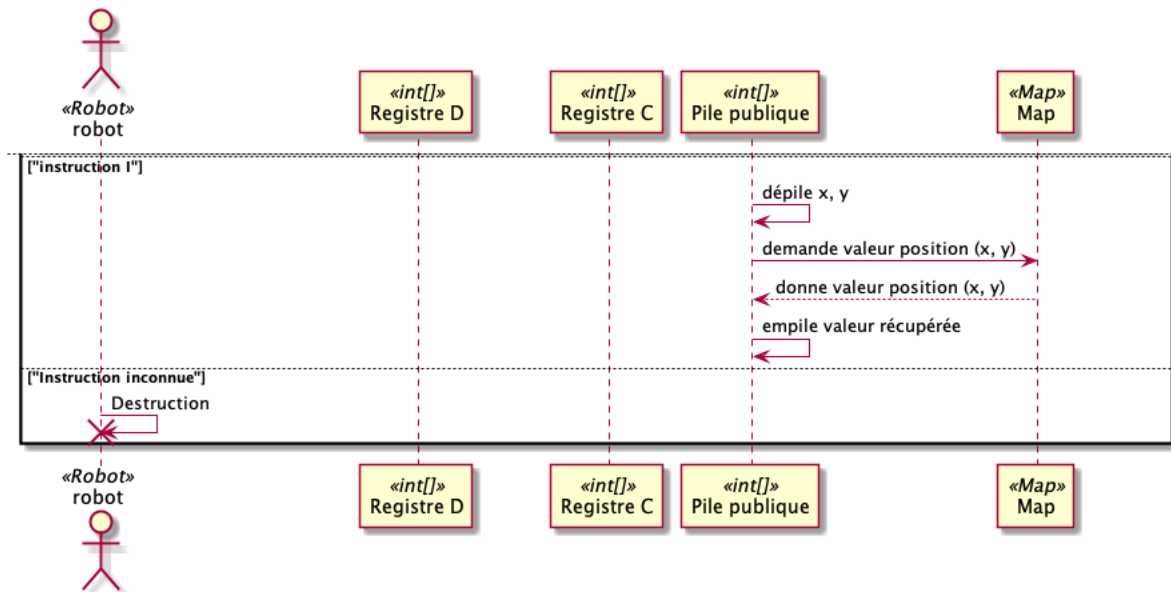
Si V = 37, il se soigne en soignant les robots à proximité. Le robot gagne autant de points d'énergie que de robots présente à côté de lui et, chaque robot à côté gagne 2 points d'énergie.

Si V = 38, le robot détruit les murs à côté de lui, il perd autant de point d'énergie que de mur à côté de lui. S'il n'a plus d'énergie, le robot est détruit.

Diagramme séquence







Pour le digramme séquence, j'ai représenté la séquence pour les différentes instructions du robot pour avoir une vision assez large du fonctionnement du robot et des interactions avec les différentes piles.