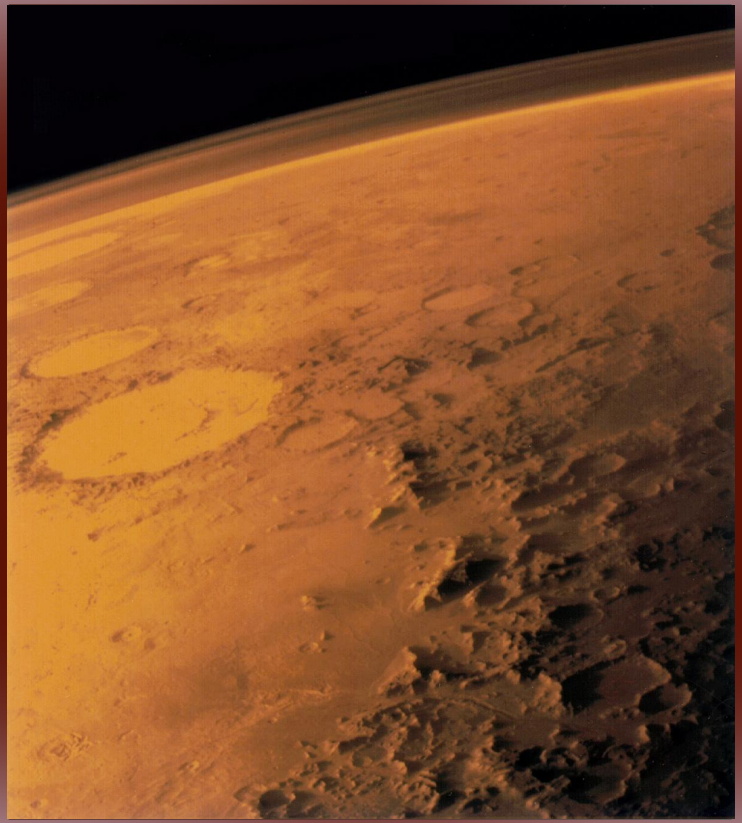
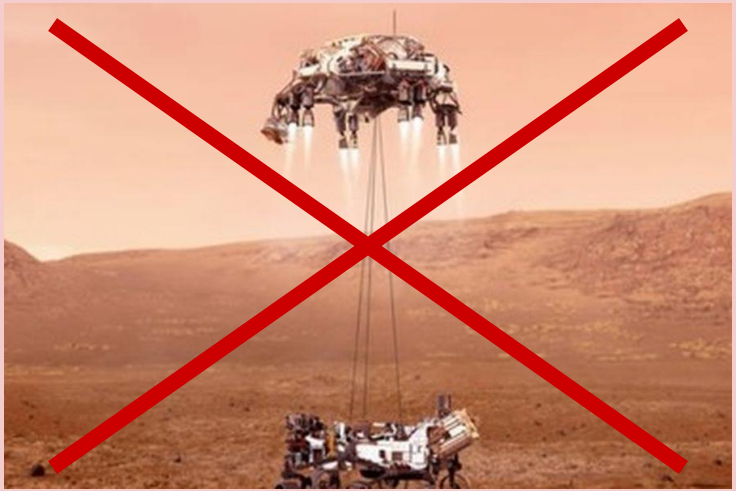
Projet conception : robot martien.



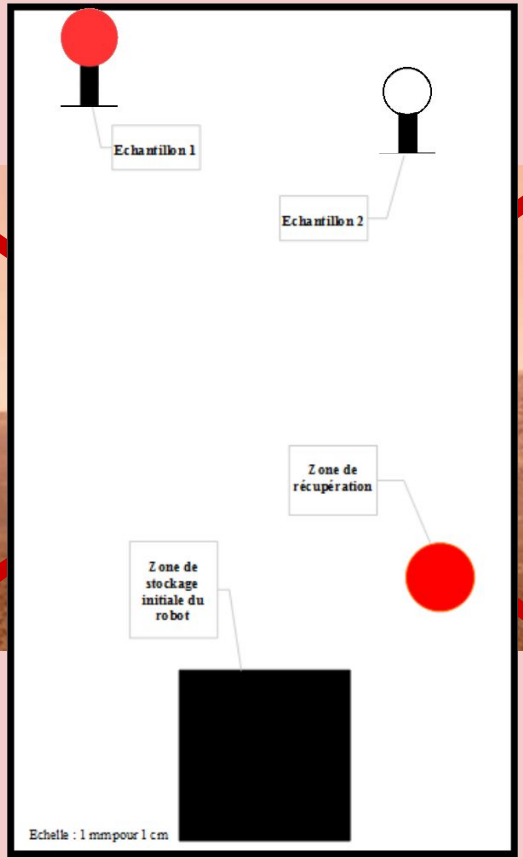
Contexte du projet.

Contexte du projet.

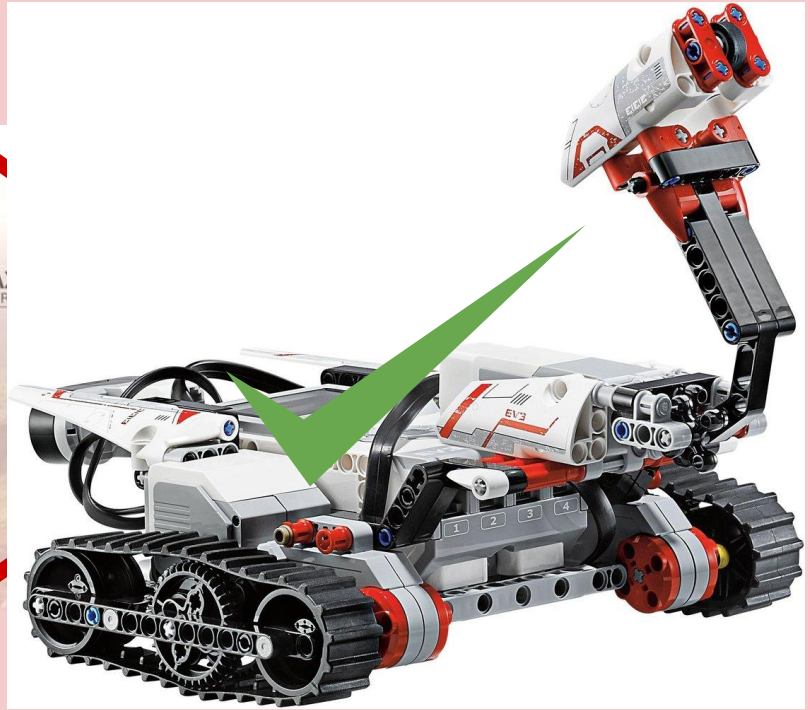
Contexte du projet.

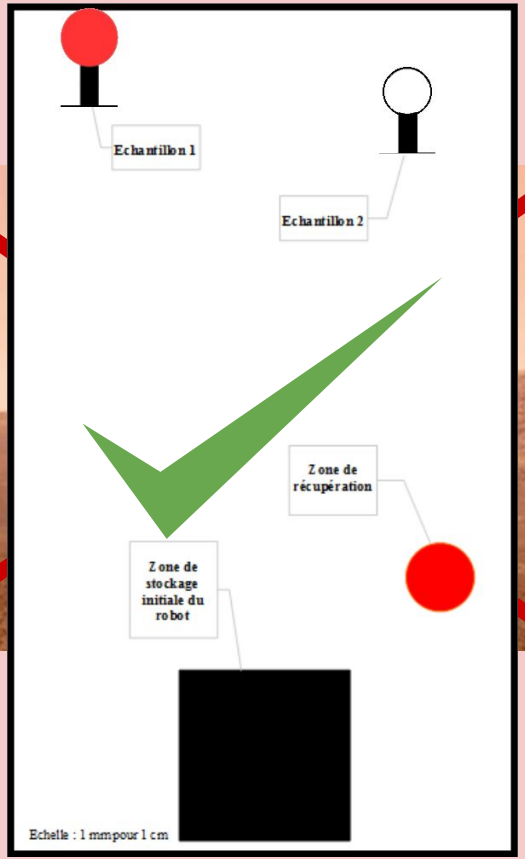


Contexte du projet.



Contexte du projet.





Démarche suivie.

Démarche suivie.

- inspiration d'un modèle physique : le track3r

Démarche suivie.

- [ ] inspiration d'un modèle physique : le track3r
- [ ] modification de la structure *on the fly*

Démarche suivie.

- [ ] inspiration d'un modèle physique : le track3r
- [ ] modification de la structure *on the fly*
- [ ] familiarisation avec la bibliothèque leJOS

Démarche suivie.

- [ ] inspiration d'un modèle physique : le track3r
- [ ] modification de la structure *on the fly*
- [ ] familiarisation avec la bibliothèque leJOS
- [ ] développement en parallèle d'algorithmes indépendants

Démarche suivie.

- [ ] inspiration d'un modèle physique : le track3r
- [ ] modification de la structure *on the fly*
- [ ] familiarisation avec la bibliothèque leJOS
- [ ] développement en parallèle d'algorithmes indépendants
- [ ] tests unitaires des capteurs et des moteurs
- [ ] tests unitaires des algorithmes
- [ ] tests grandeur nature

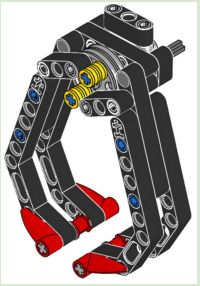
Des résultats.

Des résultats.

- inspiration d’un modèle physique : le track3r

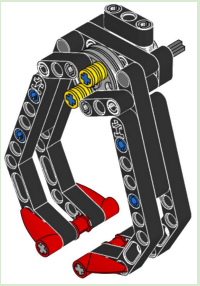
Des résultats.

- inspiration d’un modèle physique : le track3r
- modification de la structure *on the fly*



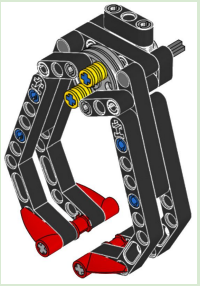
Des résultats.

- inspiration d'un modèle physique : le track3r
- modification de la structure *on the fly*
- familiarisation avec la bibliothèque leJOS



Des résultats.

- inspiration d'un modèle physique : le track3r
- modification de la structure *on the fly*
- familiarisation avec la bibliothèque leJOS
- développement en parallèle d'algorithmes indépendants



```
tant que (waypoint_index < path.length-1) faire
    rotation vers le point de passage path[waypoint_index+1] non bloquante
    tant que (rotation en cours) faire
        distance <-- mesure ultrasonique
        // calcule des coordonnées de l'objet avec la position du rover et la distance à l'objet
        objet <-- object_from_distance
        si (objet dans la zone) et (objet pas encore vu) et (objet pas dans la zone de récup) alors
            sortir de explore pour appeler harvest
        fin si
    fin tant que

    translation vers le point de passage path[waypoint_index+1] non bloquante
    tant que (translation en cours) faire
        distance <-- mesure ultrasonique
        objet <-- object_from_distance
        si (objet dans la zone) et (objet pas encore vu) et (objet pas dans la zone de récup) alors
            sortir de explore pour appeler harvest
        fin si
    fin tant que
    waypoint_index <-- waypoint_index + 1
fin tant que
```

explore

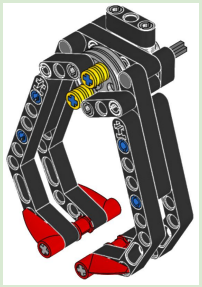
```
    fin si
fin tant que

// on est en face de l'échantillon
noter la position de l'échantillon
// pour ne plus le récolter si le rover le rencontre à nouveau
ouvrir la pince
avancer de (distance - distance entre le centre de rotation et la pince)
// pour aligner la pince et l'échantillon
fermer la pince
rotation pour [illegible] zone de récupération
translation pour approcher la zone de récupération
ouvrir la pince // pour lâcher l'échantillon
reculer
fermer la pince
```

harvest

Des résultats.

- [x] inspiration d'un modèle physique : le track3r
- [x] modification de la structure *on the fly*
- [x] familiarisation avec la bibliothèque leJOS
- [x] développement en parallèle d'algorithmes indépendants
- [~] tests unitaires des capteurs et des moteurs



```
tant que (waypoint_index < path.length-1) faire
    rotation vers le point de passage path[waypoint_index+1] non bloquante
    tant que (rotation en cours) faire
        distance <-- mesure ultrasonique
        // calcule des coordonnées de l'objet avec la position du rover et la distance à l'objet
        objet <-- object_from_distance
        si (objet dans la zone) et (objet pas encore vu) et (objet pas dans la zone de récup) alors
            sortir de explore pour appeler harvest
        fin si
    fin tant que

    translation vers le point de passage path[waypoint_index+1] non bloquante
    tant que (translation en cours) faire
        distance <-- mesure ultrasonique
        objet <-- object_from_distance
        si (objet dans la zone) et (objet pas encore vu) et (objet pas dans la zone de récup) alors
            sortir de explore pour appeler harvest
        fin si
    fin tant que
    waypoint_index <-- waypoint_index + 1
fin tant que
```

explore

```
    fin si
fin tant que

// on est en face de l'échantillon
noter la position de l'échantillon
// pour ne plus le récolter si le rover le rencontre à nouveau
ouvrir la pince
avancer de (distance - distance entre le centre de rotation et la pince)
// pour aligner la pince et l'échantillon
fermer la pi
rotation po                    zone de
translation pour approcher la zone de récupération
ouvrir la pince // pour lâcher l'échantillon
reculer
fermer la pince
```

harvest

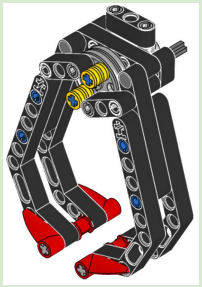
Constructeurs

Navigator

travel *rotate*

Des résultats.

- [x] inspiration d'un modèle physique : le track3r
- [x] modification de la structure *on the fly*
- [x] familiarisation avec la bibliothèque leJOS
- [x] développement en parallèle d'algorithmes indépendants
- [~] tests unitaires des capteurs et des moteurs
- [ ] tests unitaires des algorithmes



```
tant que (waypoint_index < path.length-1) faire
    rotation vers le point de passage path[waypoint_index+1] non bloquante
    tant que (rotation en cours) faire
        distance <-- mesure ultrasonique
        // calcule des coordonnées de l'objet avec la position du rover et la distance à l'objet
        objet <-- object_from_distance
        si (objet dans la zone) et (objet pas encore vu) et (objet pas dans la zone de récup) alors
            sortir de explore pour appeler harvest
        fin si
    fin tant que

    translation vers le point de passage path[waypoint_index+1] non bloquante
    tant que (translation en cours) faire
        distance <-- mesure ultrasonique
        objet <-- object_from_distance
        si (objet dans la zone) et (objet pas encore vu) et (objet pas dans la zone de récup) alors
            sortir de explore pour appeler harvest
        fin si
    fin tant que
    waypoint_index <-- waypoint_index + 1
fin tant que
```

explore

```
    fin si
fin tant que

// on est en face de l'échantillon
noter la position de l'échantillon
// pour ne plus le récolter si le rover le rencontre à nouveau
ouvrir la pince
avancer de (distance - distance entre le centre de rotation et la pince)
// pour aligner la pince et l'échantillon
fermer la pi
rotation po                    zone de
translation pour approcher la zone de récupération
ouvrir la pince // pour lâcher l'échantillon
reculer
fermer la pince
```

harvest

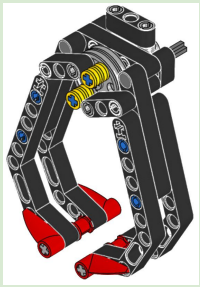
Constructeurs

Navigator

travel *rotate*

Des résultats.

- ☑ inspiration d'un modèle physique : le track3r
- ☑ modification de la structure *on the fly*
- ☑ familiarisation avec la bibliothèque leJOS
- ☑ développement en parallèle d'algorithmes indépendants
- ~ tests unitaires des capteurs et des moteurs
- ☒ tests unitaires des algorithmes
- ☒ tests grandeur nature



```
tant que (waypoint_index < path.length-1) faire
    rotation vers le point de passage path[waypoint_index+1] non bloquante
    tant que (rotation en cours) faire
        distance <-- mesure ultrasonique
        // calcule des coordonnées de l'objet avec la position du rover et la distance à l'objet
        objet <-- object_from_distance
        si (objet dans la zone) et (objet pas encore vu) et (objet pas dans la zone de récup) alors
            sortir de explore pour appeler harvest
        fin si
    fin tant que

    translation vers le point de passage path[waypoint_index+1] non bloquante
    tant que (translation en cours) faire
        distance <-- mesure ultrasonique
        objet <-- object_from_distance
        si (objet dans la zone) et (objet pas encore vu) et (objet pas dans la zone de récup) alors
            sortir de explore pour appeler harvest
        fin si
    fin tant que
    waypoint_index <-- waypoint_index + 1
fin tant que
```

explore

```
    fin si
fin tant que

// on est en face de l'échantillon
noter la position de l'échantillon
// pour ne plus le récolter si le rover le rencontre à nouveau
ouvrir la pince
avancer de (distance - distance entre le centre de rotation et la pince)
// pour aligner la pince et l'échantillon
fermer la pince
rotation pour [illegible] zone de récupération
translation pour approcher la zone de récupération
ouvrir la pince // pour lâcher l'échantillon
reculer
fermer la pince
```

harvest

Constructeurs

Navigator

travel *rotate*

Conclusion et enseignements.

Conclusion et enseignements.

construire

Conclusion et enseignements.

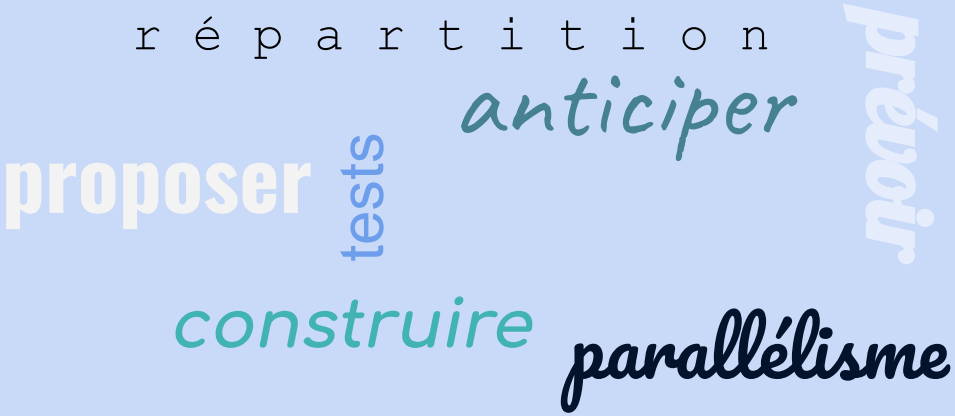
anticiper

prévoir

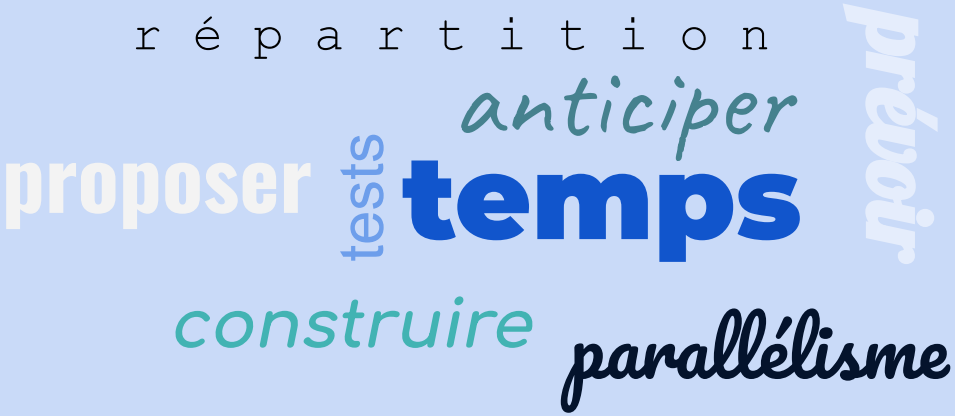
construire

Conclusion et enseignements.

Conclusion et enseignements.



Conclusion et enseignements.



FIN