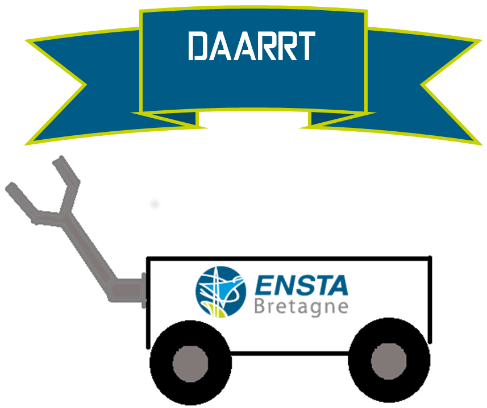
Datasheet - Simulateur

# Description

Le simulateur du DAARRT, développé par l’équipe réalisant le projet, est un outil mis à disposition des utilisateurs du robot pour tester leurs cerveaux et donc vérifier la bonne marche de leurs programmes. Les vitesses et dimensions sont donc fictives.

Le simulateur reste une approximation de la réalité et ne saurait donc représenter tous les aspects d’une mission réelle. Les éléments suivants ne sont donc **PAS** pris en compte par le simulateur :

* Préhension d’objet avec le bras
* Déviation naturelle du robot sur une ligne droite
* Gestion de la webcam
* Gestion de l’aspect sonore

L’un des principaux apports du simulateur du DAARRT par rapport au simulateur du JOG est sa modularité et la facilité de modifier certains éléments. Ces modifications éventuelles, comme la création d’un nouveau monde, sont détaillées dans la dernière partie.

# Fonctionnement du simulateur

Le simulateur est codé en langage python, le même langage que celui utilisé pour coder le cerveau du DAARRT. Lors du lancement d'un programme python celui-ci instancie un objet de type daarrt, cet objet, selon qu'il est lancé sur ordinateur ou sur le robot, il détecte ou non un fichier de config et fait les bons imports.

Le simulateur est programmé pour tourner jusqu'à fin du programme ou jusqu'à collision avec un obstacle. Cette information est retournée sous forme texte dans la console de l'utilisateur.

L'outil pygame est principalement utilisé pour proposer un rendu 2D du robot.

La fenêtre à droite propose différentes vues du bras robotique et d'autres éléments.

# S’approprier le simulateur

Le principal fichier que l'utilisateur peut être amené à modifier est le fichier *constantes.py*  où se trouve toutes les valeurs susceptibles de modifier le comportement de la simulation.

L'utilisateur peut aussi allez modifier le fichier world.txt. Celui gère la génération du monde. L'utilisateur peut rajouter des O afin de rajouter des obstacles sur le parcours. Le L sur l'avant dernière ligne ne doit pas être retiré, c'est un raccourci de code utile pour la génération pygame du monde. Une lettre D désigne l'emplacement du centre du robot sur le monde. En son absence le robot sera positionné en (250,250).

Le repère du simulateur est le suivant :

# Comparaison simulateur-composants

|  |  |  |
| --- | --- | --- |
| **Composant** | **Fonctionnalités réelles** | **Fonctionnalités simulées et limitations** |
| Sonars | Détection impossible en dessous de 5 cm | Ok |
| Détection impossible au-delà de x cm | Ok |
|  |  |
| Centrale d’attitude | 9 axes | Cap uniquement, référentiel du simulateur dans la datasheet |
|  |  |
|  |  |
|  |  |
| Carte T’Rex | Ecriture sur le bus I2C | OK (complète) |
| Lecture sur le bus I2C | OK mais seules valeurs non nulles :  Vitesses, Encodeurs |
|  |  |
| Audio | Acquisition de son | Non |
| Emission de son | Non |
| Bras | 2 degrés de liberté | Ok |
| Préhension d’objet | Non |
|  | Attention le bras compte constamment dans le corps physique du robot, qu’il soit levé ou non. En Simulation le contact avec un mur peut donc se faire plus « tôt » qu’il ne se ferait dans la réalité |
|  |  |  |