Robot - TP1 - Simu

Installation

Depuis https://gitlab.cri.epita.fr/jeremie.graulle/ssie-robot-simulator créez un “fork” personnel privé.

Ajoutez votre binome et “jeremie.graulle” en “maintainer”.

Créez une branche tp1 pour commiter vos modifications.

Compilez le projet en suivant le README.txt soit avec VS code soit en ligne de commande.

Dans les 2 cas tous les fichiers générés devront être placé dans un sous dossier et ce dossier devra être ajouté dans le fichier `.gitignore` pour ne pas commiter les fichiers générés.

Attention: Une partie de la note prendra en compte qu’aucun fichier généré n’est commité.

Etapes

Recherchez les “TODO” suivant et suivez les instructions:

1. Dans `Robot::update()` recherchez “TODO Step1”
   - Attention aux unités angulaires de `setRotation()`
   - Là où c’est pertinent, utilisez des constantes (voir cppreference)
   - Si vous ajoutez des coefficients, faites attention de bien respecter le comportement suivant : Si on appuit sur les touches ‘a’ ou ‘e’ on veut qu’une seule roue tourne, l’autre roue ne doit pas bouger, le centre de rotation du robot doit se situer sur le bord du robot. Essayez de trouver les formules par vous même (par calcul ou sur Internet) mais si vous ne trouvez pas voici quelques exemples : [1] [2] [3] [4]
2. Dans `LineTrackSensor::update()` recherchez “TODO Step1”
3. Dans `IrProximitySensor::update()` recherchez “TODO Step1”
   - Avant de partir sur des calculs complexes, essayez de bien comprendre ce que fait : `_line[1].position = worldTranform.transformPoint(sf::Vector2f(_distanceDetected, 0.0));`
4. Dans `SwitchSensor::update()` recherchez “TODO Step1”
5. Dans `Robot::update()` recherchez “TODO Step2”
6. Dans `IrProximitySensor::update()` recherchez “TODO Step2”
7. Dans `EncoderWheelSensor::update()` recherchez “TODO Step1”
   - C’est une barrière infrarouge qui permet de détecter un obstacle entre la partie émetteur et récepteur
   - Elle est utilisée avec une roue en plastique contenant 20 fentes permettant de détecter des 20eme de tour
   - Cette roue est directement mise sur le même arbre que celui de la roue permettant donc de détecter des 20eme de tour de chaque roue
   - Une manière de quantifier cette information est d’incrémenter un compteur ici _value à chaque évènement du capteur
   - Il faut donc ici calculer la distance parcourue par la roue dans le pas de simulation et mettre à jour la valeur intermédiaire _distance
   - Puis de mettre à jour la valeur _value par rapport au nombre de 20eme de roue que vous pensez que le capteur a détecté dans ce pas de simulation
   - Ce capteur n’a aucune influence sur le déplacement du robot avec les touches, le seul moyen pour l’instant de voir son comportement est d’afficher en console les valeurs (n’oubliez pas d’enlever ce code de debug pour le commit)

Vous pouvez pousser vos modifications de la branche TP1 sur le gitlab autant de fois que vous voulez (pour ne pas perdre vos modifications).

Une fois fini, créez une “merge request” depuis la branche `tp1` vers le `main` dans votre “fork” personnel, puis ajoutez jeremie.graulle en reviewer de cette MR.