

## DOCUMENTATION DEVELOPPEUR SIMULATION VREP-ROS

La modélisation VREP nommée « Simulation.ttt » est celle utilisée lors d'une simulation. Elle représente la cellule Montrac que nous avons développée, sans les navettes qui y circulent. En effet, lors d'une simulation, celles-ci sont chargées dans la simulation par le code C++. Pour y accéder dans VREP, il faut les placer manuellement via le model browser : elles sont disponibles dans le dossier « CelluleFlexible » et il suffit alors de les glisser dans la fenêtre de modélisation.

Concernant les différentes parties de la modélisation, voici l'explication de leurs créations et propriétés (description de haut en bas dans la scène hiérarchie). Les propriétés principales sont « responsable » (physique), « visible » (visible), et « dynamic » (dynamique). Elles sont disponibles dans les différents onglets de Tools->Scene object properties lorsqu'un objet est sélectionné.

### LA CELLULE ET SON ENVIRONNEMENT

La DefaultCamera : caméra présente par défaut qui permet la visualisation de la scène.

Les DefaultLightX (ex : DefaultLightA) : lumière d'ambiance pour la visualisation de la simulation.

Les Machines (ex : Machine droite atelier bas droite) : machine d'atelier ou la navette peut avoir à subir des opérations (uniquement visible).

Le Montrac : ensemble de formes écrivant « MONTRAC » sur le mur de la simulation (uniquement visible).

Les Rail jonction (ex : Rail jonction droite bas droite) : petite partie de rail de la cellule (visible et physique).

Les Socle aiguillage (ex : Socle aiguillage A1) : socle sur lequel repose un aiguillage (uniquement visible).

Le Vision sensor : caméra permettant de voir la simulation lorsqu'elle est lancée via un code extérieur. Dans son script (petit icône de page à côté du nom), on publie les images qu'elle capture dans un topic auquel on s'abonnera pour les récupérer.

Les Rails alimentés (ex : Rail droite bas) : partie physique de rail de la cellule à laquelle est attachée une partie uniquement visible correspondant à l'alimentation de la navette (cliquer sur l'icône « + » à côté du nom pour voir l'arborescence).

Les Window140cm (ex : window140cm) : image visible de fenêtre entourant la cellule complète.

Les Pieds (ex : Pieds milieu droite) : image visible des pieds et supports de la cellule.

Le ResizableFloor 5\_25 : sol physique sur lequel repose la cellule.

Le StationSensor : ensemble des capteurs situés sur les machines d'atelier (pour détecter les navettes). Dans son script, on publie les états des capteurs via un unique entier (chaque capteur activé renvoie 0 ou  $2^{(\text{numéro du capteur}-1)}$ ). Cet entier est publié dans le topic "vrep/StationSensor"

Le SwitchSensor : ensemble des capteurs situés sur les aiguillages (pour connaître leur positions). Même principe que pour son script que celui dans StationSensor, en publiant l'entier dans le topic "vrep/SwitchSensor".

Le RailSensor : ensemble des capteurs situés le long des rails de la cellule (pour détecter les navettes). Même principe que pour son script que celui dans StationSensor, en publiant l'entier dans le topic "vrep/RailSensor".

Le Aiguillages : ensemble des aiguillages, chacun composé d'un socle fixe, d'un joint faisant tourné un socle mobile, et du rail sur lequel circule la navette. Dans son script, on s'abonne aux topics `"/commande_locale/SwitchControllerRight"`, `"/commande_locale/SwitchControllerLeft"` et `"/commande_locale/SwitchControllerLock"` qui reçoivent des entiers de 32 bits publié par un code C++ correspondant respectivement aux états des actionneur droite, actionneur gauche et verrouillage/déverrouillage.

Le Stop : ensemble des ergots métalliques correspondant aux endroits où la navette peut avoir à s'arrêter. Chaque stop est composé d'un 1<sup>er</sup> ergot vu par la navette qui la fait ralentir, et d'un joint sur lequel est fixé un 2<sup>ème</sup> ergot qui la fait s'arrêter si elle le voit, et repartir quand elle ne le voit plus. Dans son script, on s'abonne aux topics `"/commande_locale/StopController"` et `"/commande_locale/StopController"` qui reçoivent des entiers de 32 bits publié par un code C++, et ces entiers sont utilisés pour déplacer les « deuxièmes » ergots métalliques en positions stop ou continu.

## LA NAVETTE ET SON FONCTIONNEMENT

Concernant la modélisation de la navette (ex : « ShuttleA»), voici les différents éléments qui la constituent.

Front sensor: capteur de proximité de type « ultrasonic » situé à l'avant de la navette. Il permet la détection des navettes situées devant afin d'éviter toutes collisions.

Number : numéro inscrit sur le dessus de la navette. Permet aux utilisateurs de distinguer les navettes lorsqu'il y en a plusieurs dans la simulation.

PlatformUp : plate-forme supérieure de la navette. Partie physique qui est détectable uniquement par les lasers. C'est cette partie de la navette qui sera détectée lorsque celle-ci passe sur un capteur placé dans les zones de travail (là où se situent les machines).

PlatformDown : plate-forme inférieure de la navette. Partie physique mais non détectable.

BackResponder : pièce non-visible placée à l'arrière de la navette. Elle est détectable par les capteurs « ultrasonic ». C'est cette partie qui sera détectée lorsqu'une navette qui la suit se rapproche trop.

Revolute joint back : joint sur lequel est fixé le mécanisme des roues arrières. Le mécanisme est constitué d'un socle sur lequel on retrouve deux joints non-moteurs fixés à deux sphères.

Revolute joint front : joint sur lequel est fixé le mécanisme des roues avants. Le mécanisme est constitué d'un socle sur lequel on retrouve deux joints moteurs qui entraînent en rotation des sphères.

Arms : bras qui n'est ni physique ni dynamique. Utilisé uniquement pour que le design se rapproche de la navette réelle.

InductiveSensor : capteur de proximité de type « inductive ». Il détecte les ergots métalliques qui vont faire ralentir la navette.

UltrasonicSensor : capteur de proximité de type « ultrasonic ». Il détecte les ergots métalliques qui vont faire s'arrêter la navette.

SensorResponder : partie de la navette détectable avec laser. Placée sur le côté elle permet d'activer les capteurs de position situés sur les rails lors du passage de la navette.

#### Script associé à la navette :

Le script associé à chaque navette va permettre de gérer sa vitesse. Il s'abonne au topic MotorVelocityTopic puis règle la vitesse des roues de devant au lancement de la simulation.

Puis lorsque les capteurs inductif et ultrasonic détecte un slow ou un stop, la vitesse est modifiée en conséquence.