

UE 4.2 Simulation - Exercice CoppeliaSim noté - 29 Mars 2024 – 8h15 à 12h15

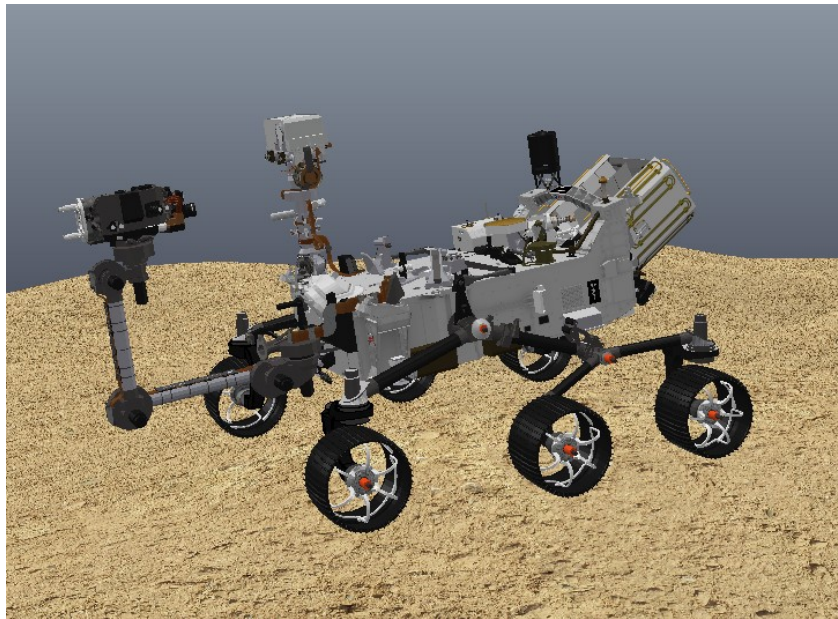
Durée : 4h

Tous documents autorisés. LLM (Chat-GPT, Claude, Gemini ...) autorisés. Possibilité de réutiliser et adapter les codes réalisés en TD.

Le rendu se fera sur le dépôt GitLab du cours UE 4.2 Simulation. En principe, au début du cours, vous m'avez déclaré "reporter" sur votre dépôt, si ce n'est pas le cas, il est temps de le faire.

Pour ce BE noté, l'idéal est de faire au moins un commit à chaque question avec un intitulé de commit contenant "eval qn" avec n le numéro de la question. Vous pouvez faire d'autres commits par la suite pour corriger ou améliorer votre travail et l'idéal est que le libellé de ces commits contienne "eval qn". Le travail rendu au dernier commit avant 12h15 sera pris en compte pour la notation.

Pour ce BE noté, il faudra créer le modèle dynamique du robot Perseverance de la NASA. La CAO très simplifiée est dans le fichier "perseverance-lab-v0-2.ttt"



Q1 (5 pts) : A l'aide de la "zmqRemoteApi", créer un programme Python qui renvoie la position du corps principal (body) et la dimension des roues. Faire un commit contenant "eval q1" dans le libellé.

Q2 (5 pts) : Réaliser un modèle dynamique simple de Perseverance avec le corps principal et les 6 roues. Chaque roue est reliée au corps principal par une liaison pivot (revolute joint). Rendre invisible le modèle dynamique et sauvegarder le robot dans un fichier nommé "perseverance-q2.ttt". Faire un commit contenant "eval q2" dans le libellé.

Q3 (5 pts) : Commander le robot en le faisant avancer en ligne droite pendant deux secondes puis en le faisant tourner sur lui-même pendant deux secondes. Vous avez le choix d'utiliser ROS2, la "zmqRemoteApi" ou un socket.

- Si vous choisissez le socket il faudra fournir les deux programmes Python : le script serveur (script dans CoppeliaSim) et le programme client (programme externe de contrôle) ou l'inverse si vous avez choisi d'exécuter le client dans CoppeliaSim.
- Si vous choisissez ROS2 il faudra fournir le script python sur CoppeliaSim et un programme extérieur de contrôle sous forme de bash avec les commandes ROS2 (ros2 topic pub ...) successives (*note : pour attendre 2 secondes la commande bash est sleep 2*).
- Si vous choisissez la "zmqRemoteApi", il faudra fournir le programme Python

Faire un commit contenant "eval q3" dans le libellé.

Q4 (5 pts) : Améliorer le modèle dynamique en ajoutant des liaisons rendant le robot Perseverance plus réaliste, en lui permettant de se déplacer en tout terrain (vous pouvez récupérer un terrain 3D dans "perseverance-lab-v0-2-desert.ttt" ou "perseverance-lab-v0-2-desert.ttt". Le résultat sera dans un fichier nommé "perseverance-q4.ttt".

Faire un commit contenant "eval q4" dans le libellé.