

MASTER PROFESSIONNEL RAIA

TRAVAUX PRATIQUES

Atelier Robots Mobiles

TPN°1 : Commande des robots mobiles et évitement d'obstacles

Date : ----- **Classe :** ----- **Durée : 3h**

	<i>Nom & Prénom :</i>	<i>AB/PR</i>	<i>Mot/Part</i>	<i>TP N°</i>	<i>Total</i>
1	-----		/10	/10	/20
2	-----		/10	/10	/20
3	-----		/10	/10	/20

Objectifs du TP :

- ✓ -----
- ✓ -----
- ✓ -----

Conditions de réalisation et moyens :

- ✓ -----
- ✓ -----
- ✓ -----

Objectifs :

- Contrôler un robot mobile en utilisant Matlab et V-REP Pro Edu (CoppeliaSim Edu).
- Maîtriser les outils de base pour contrôler un robot Mobile en utilisant Matlab et V-REP.
- Mettre en œuvre une stratégie d'évitement d'obstacles

I- Introduction :

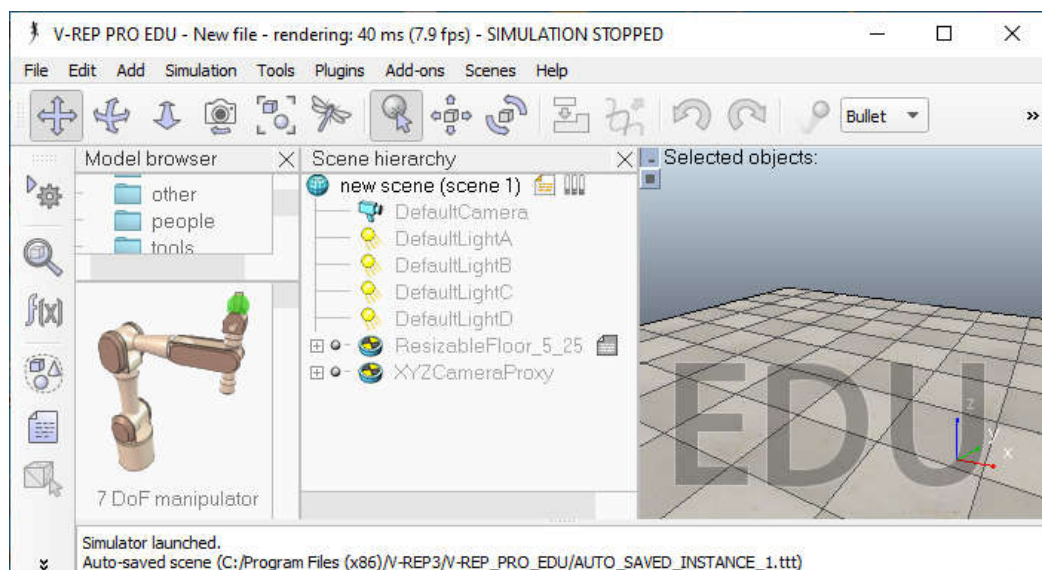
V-REP Pro Edu est un logiciel de simulation robotique 3D avancé, qui nous permet de simuler le comportement de robots dans des environnements virtuels. Développé par Coppelia Robotics [1], il est largement utilisé dans les domaines de la recherche, de l'éducation et de l'industrie pour la conception, l'automatisation, la programmation et la validation de systèmes robotiques.

Avec V-REP Pro Edu, nous pouvons créer des scénarios de simulation à l'aide d'un environnement graphique convivial. Le logiciel prend en charge une grande variété de robots, de capteurs et d'actionneurs, tels que les caméras, les capteurs de force, les actionneurs pneumatiques, les moteurs électriques et les servomoteurs, ce logiciel permet également la programmation de robots à travers différents langages de programmation tels que le Python, le C++, le Java, le Lua, le Matlab, etc.. Nous pouvons aussi définir les propriétés physiques de chaque objet dans la simulation, telles que la masse, la friction et l'inertie, et nous pouvons également ajuster les propriétés de simulation, telles que la gravité et la résolution de la collision.

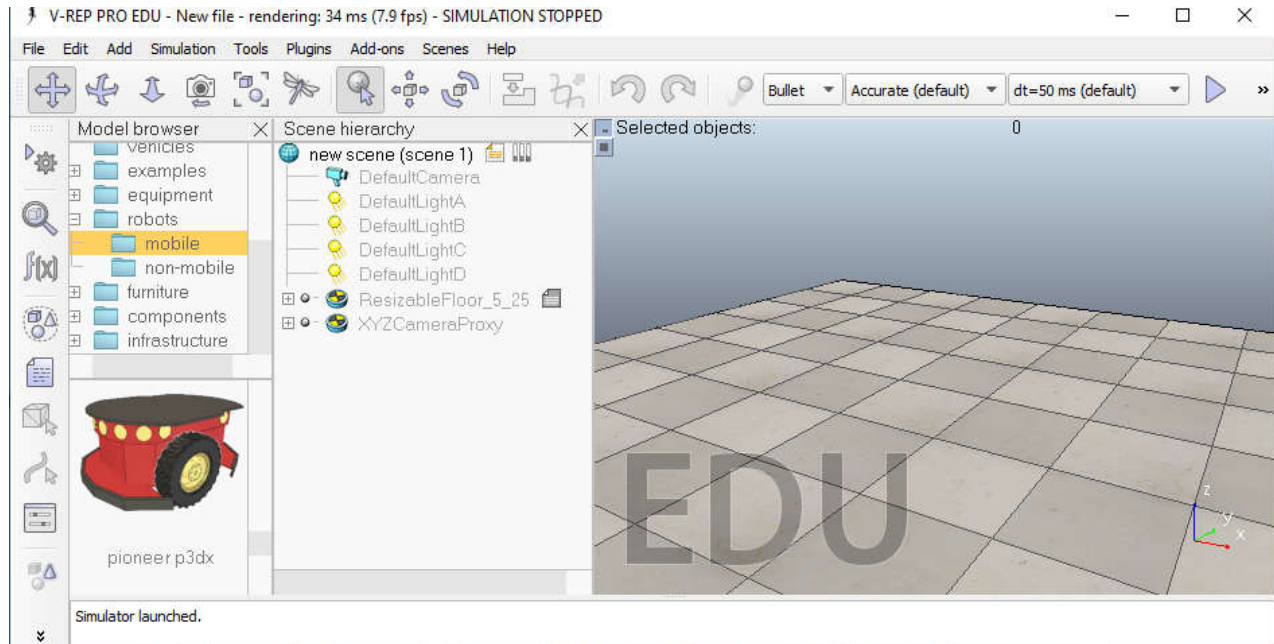
En plus de la simulation de robots, V-REP Pro Edu dispose également de fonctionnalités avancées telles que la vision par ordinateur, l'apprentissage en profondeur et la planification de trajectoire permettant la création de politiques de contrôle pour les robots. Nous pouvons également exporter les scénarios de simulation sous forme de code source pour une utilisation sur des plateformes robotiques réelles.

II- Préparation du l'environnement V-REP Pro Edu :

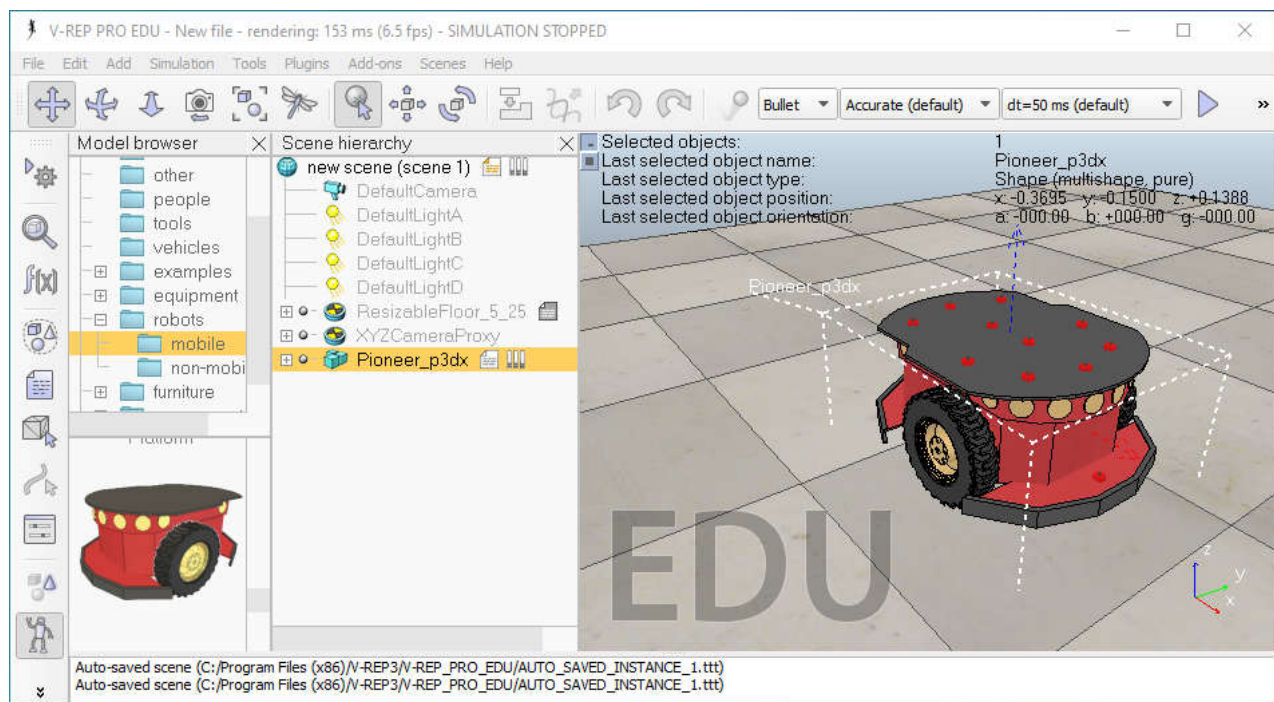
Pour commencer à travailler sur des projets de simulation robotique, ouvrir V-REP en double-cliquant sur l'icône V-REP PRO EDU. La fenêtre suivante s'affiche



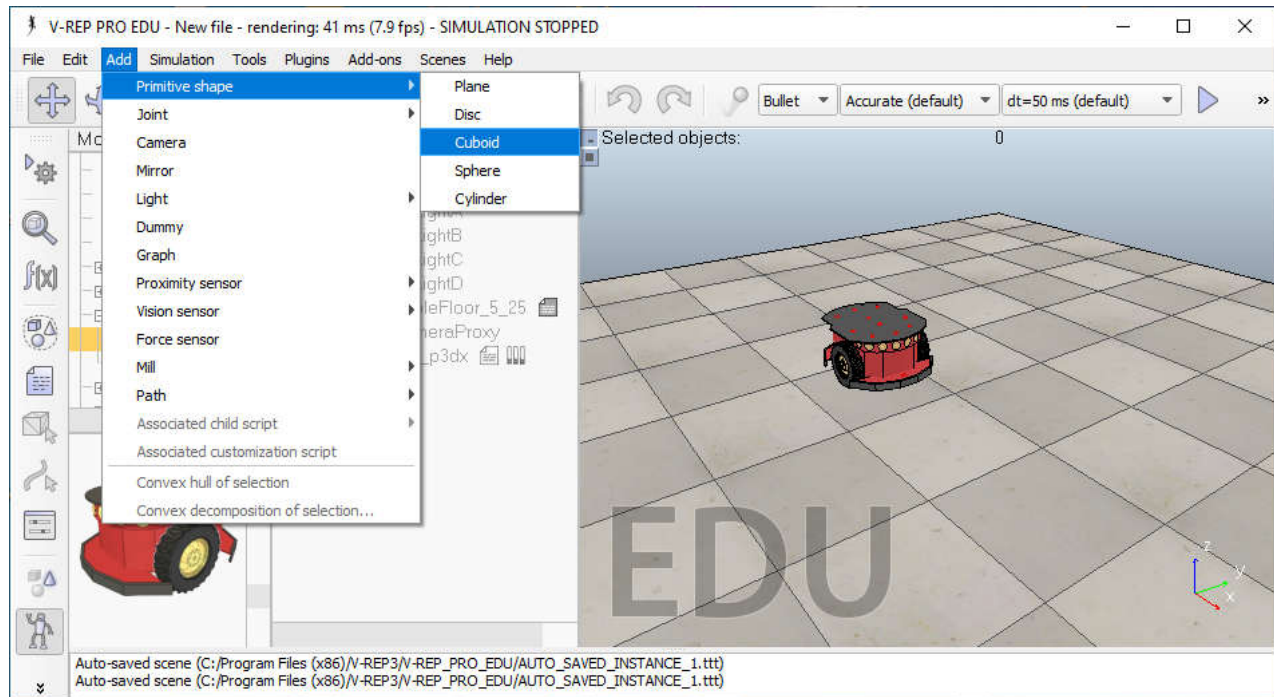
Dans l'espace du « model browser », répertoire « robots », sous-répertoire « mobile » choisir le robot « Pioneer P3dx ».



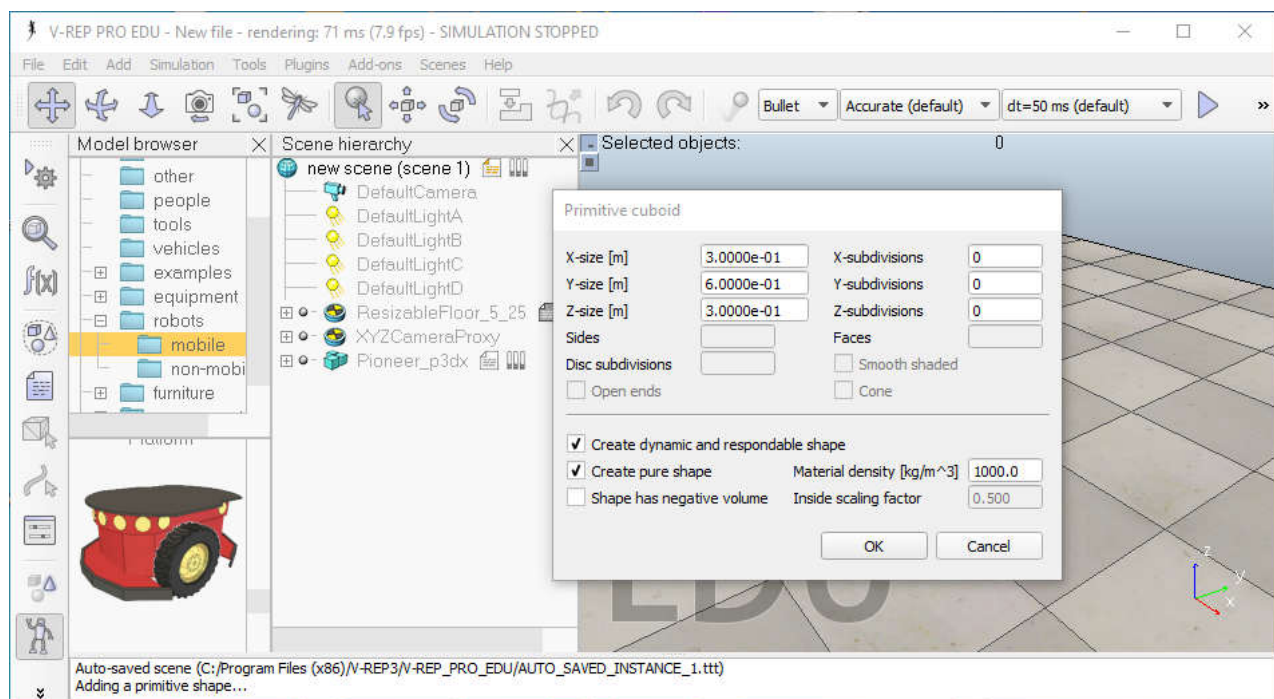
Déposer le robot mobile dans la scène



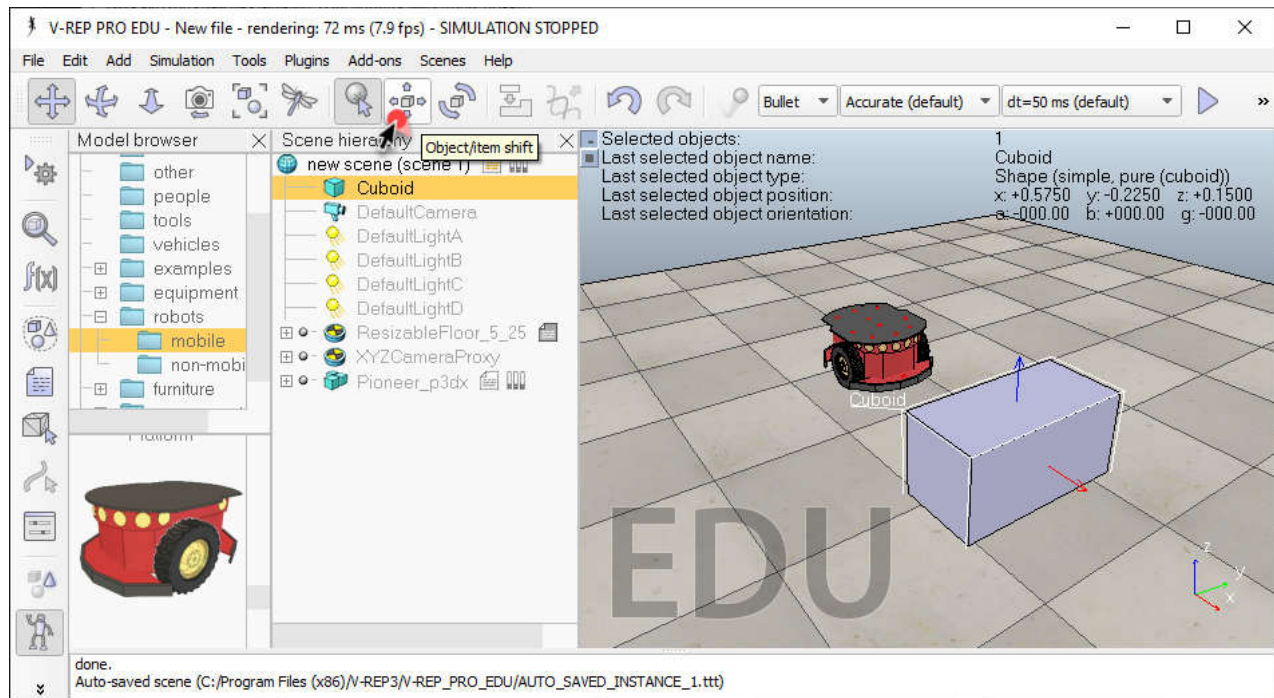
Ajouter à la scène un simple « Cuboid »



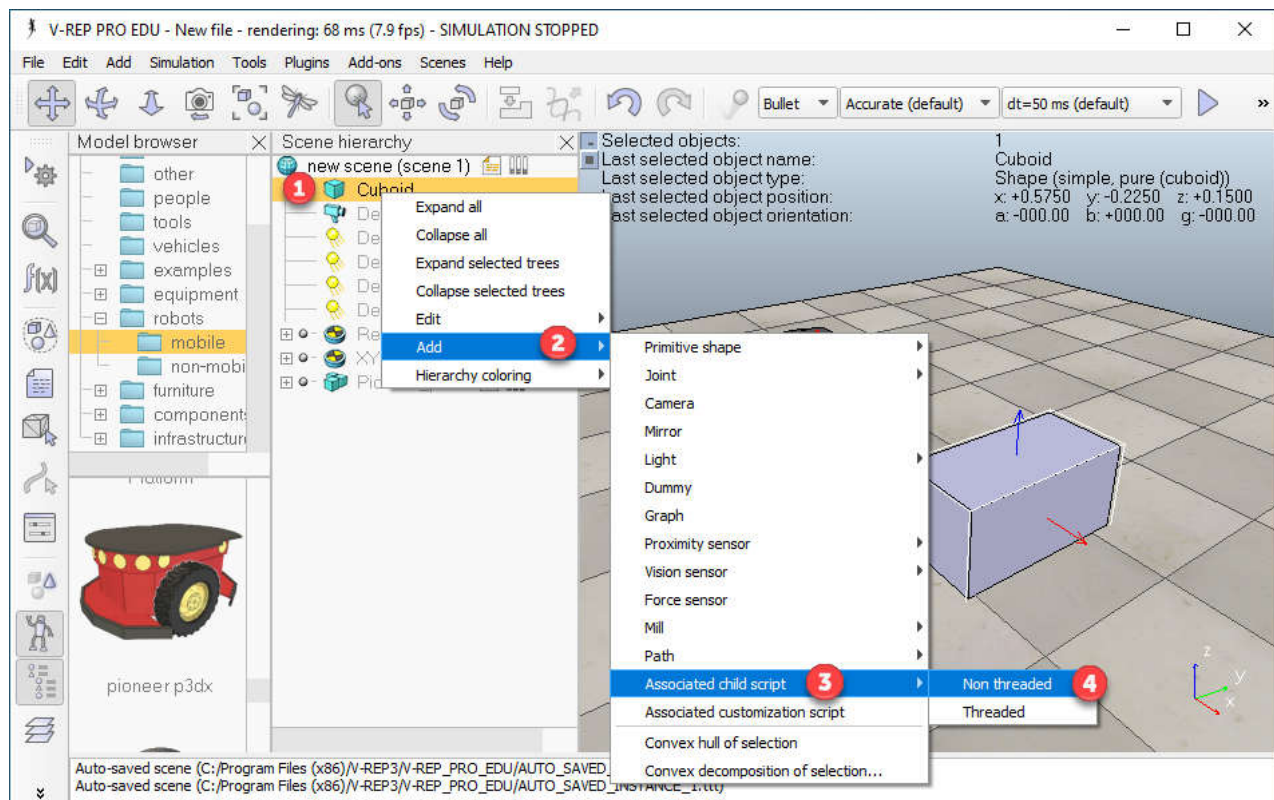
Dans la fenêtre qui s'affiche, paramétrer X, Y et Z-size[m] : (0.3 ;0.6 ;0.3)m puis appuyer sur ok



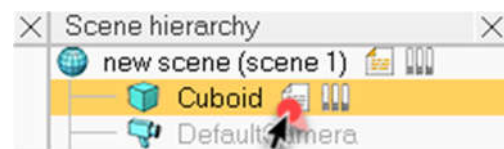
Pour changer la position des objets dans la scène : cliquer sur l'icône « Object/item shift », indiqué par la flèche, puis sélectionner l'élément à déplacer.

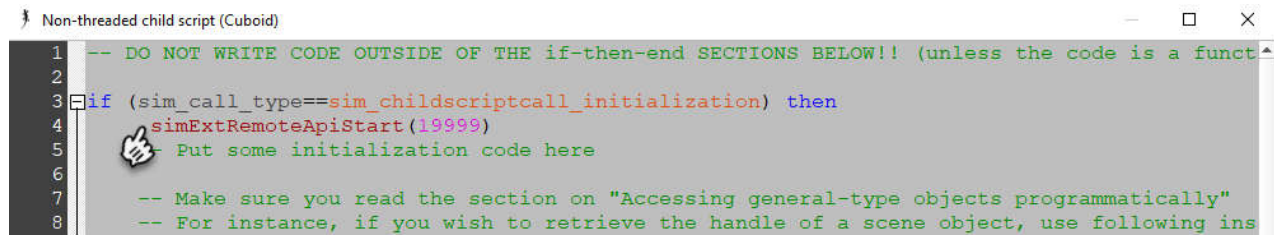


Dans la « scene hierarchy », cliquer sur « Cuboid », « Add », Associated child script », puis sélectionner « non threaded »



Cliquer sur le symbole d'une feuille qui est attaché à « Cuboid », la fenêtre du script s'affiche et on va l'ajouter la ligne suivante pour fixer le port de communication entre Matlab et V-REP : `simExtRemoteApiStart (19999)`





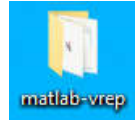
```

1  -- DO NOT WRITE CODE OUTSIDE OF THE if-then-end SECTIONS BELOW!! (unless the code is a funct
2
3  if (sim_call_type==sim_childscriptcall_initialization) then
4      simExtRemoteApiStart(19999)
5      % Put some initialization code here
6
7      -- Make sure you read the section on "Accessing general-type objects programmatically"
8      -- For instance, if you wish to retrieve the handle of a scene object, use following ins


```

Fermer le script, puis créer un nouveau répertoire sous le nom de « **matlab-vrep** ».

Enregistrer votre scène dans le répertoire crée en portant le même nom « **matlab-vrep** »



⇒ Consulter le chemin suivant :

 **matlab-vrep.ttt**

C:\Program Files (x86)\V-REP3\V-REP_PRO_EDU\programming\remoteApiBindings\matlab\matlab

Copier les scripts portant les noms suivant dans le répertoire « **matlab-vrep** ».

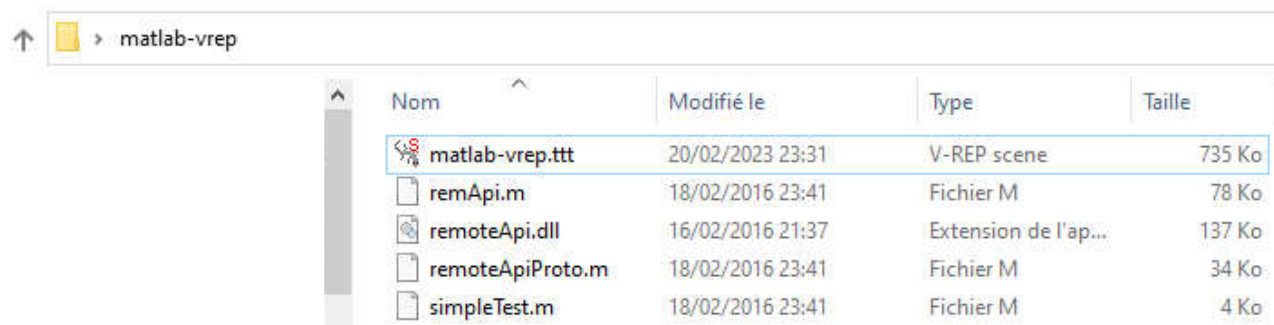
« remApi.m », « remoteApiProto.m », « simpleTest.m »

⇒ Consulter le chemin suivant :

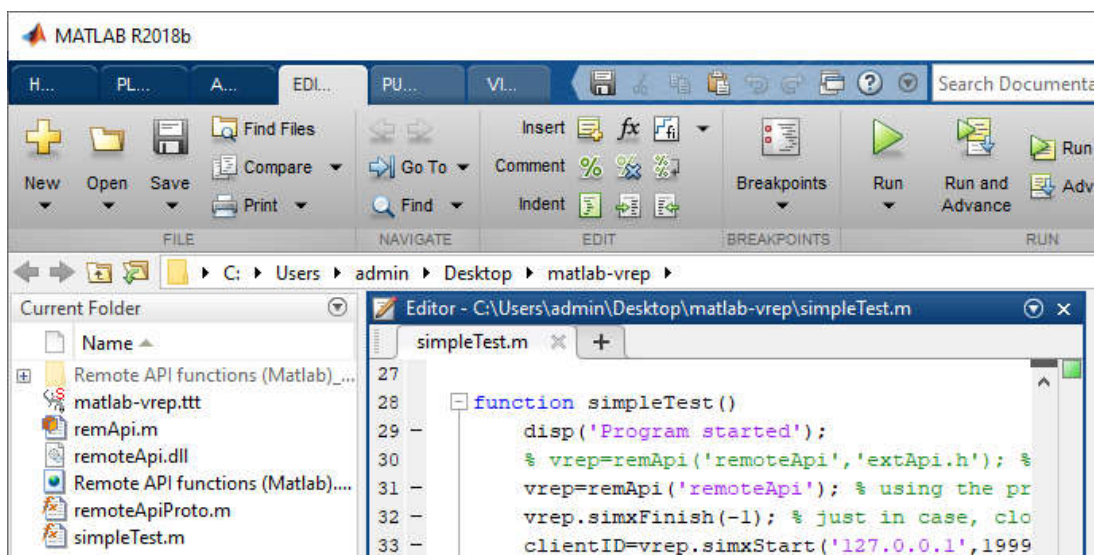
C:\Program Files (x86)\V-REP3\V-REP_PRO_EDU\programming\remoteApiBindings\lib\lib\64Bit

Copier le fichier « remoteApi.dll » dans le répertoire « **matlab-vrep** ».

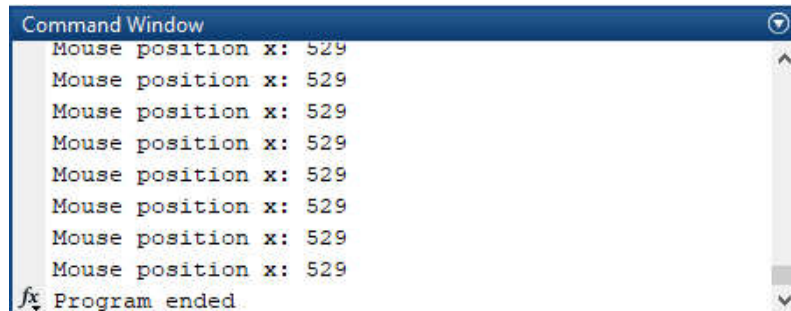
Vérifier que vous avez recopié tous les fichiers nécessaires à la simulation dans le répertoire « **matlab-vrep** ».



Lancer le logiciel Matlab et ajouter le schéma du répertoire « **matlab-vrep** » au workspace.



Ouvrir le script « simpleTest.m », cliquer sur le bouton de simulation du V-REP puis celle du Matlab. Si vous déplacer la souris dans la scène, où se trouve le robot, Matlab affichera dans la commande Windows la position en temps réel de la souris, qui va indiquer que la connexion est établie.



III- Contrôle du robot via Matlab et V-REP :

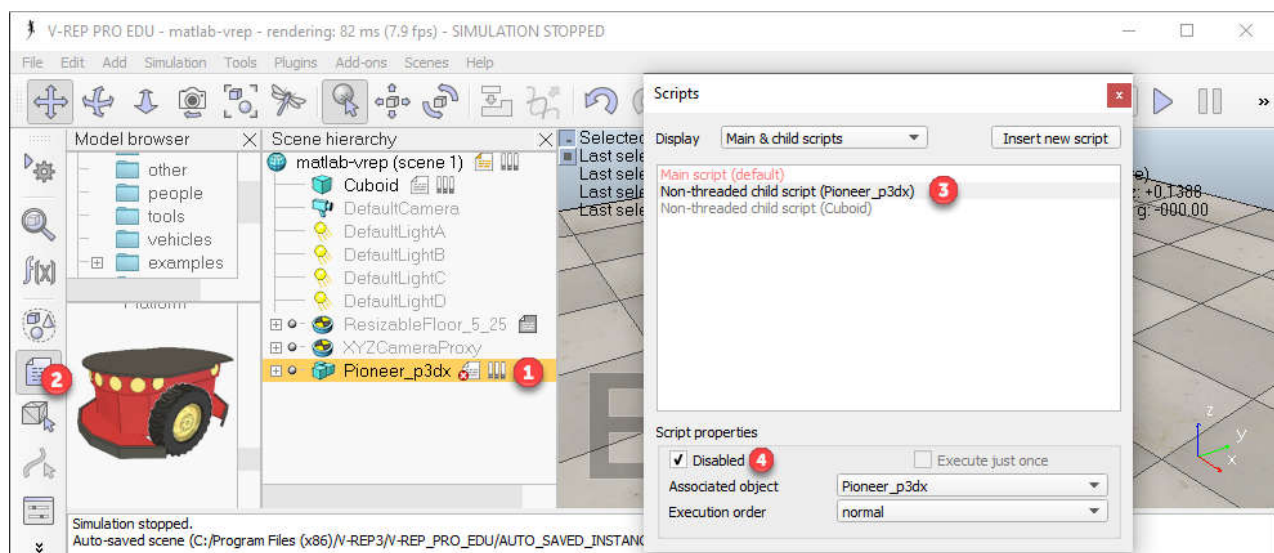
Les fonctions « Legacy remote API functions (Matlab) » dans V-REP permettent aux utilisateurs de contrôler à distance une simulation V-REP depuis MATLAB.

<https://www.coppeliarobotics.com/helpFiles/en/remoteApiFunctionsMatlab.htm>

1- Commande d'un Moteur du robot « Pioneer_p3dx »

Pour contrôler le robot depuis Matlab, il faut désactiver la simulation depuis V-REP :

Dans la « Scene hierarchy », sélectionner « Pioneer_p3dx », « Script ». Quand une fenêtre s'affiche, cliquer sur « non-threaded child script (Pioneer_p3dx) » puis cocher « Disabled » ainsi le robot sera contrôlé que par Matlab seulement. Si vous lancez la simulation via V-REP le robot ne bouge pas.



Dans Matlab, créer un nouveau script et écrire le code suivant et l'enregistrer :

```
vrep=remApi('remoteApi');
vrep.simxFinish(-1);
clientID=vrep.simxStart('127.0.0.1',19999,true,true,5000,5);
if (clientID>-1)
    disp('Connected')
```



```

[returnCode, left_motor]=vrep.simxGetObjectHandle(clientID, 'Pioneer_p3dx_leftMotor',vrep.simx_opmode_blocking)

[returnCode]=vrep.simxSetJointTargetVelocity(clientID, left_motor,
3,vrep.simx_opmode_blocking)

pause(5)

[returnCode]=vrep.simxSetJointTargetVelocity(clientID, left_motor,
0,vrep.simx_opmode_blocking)

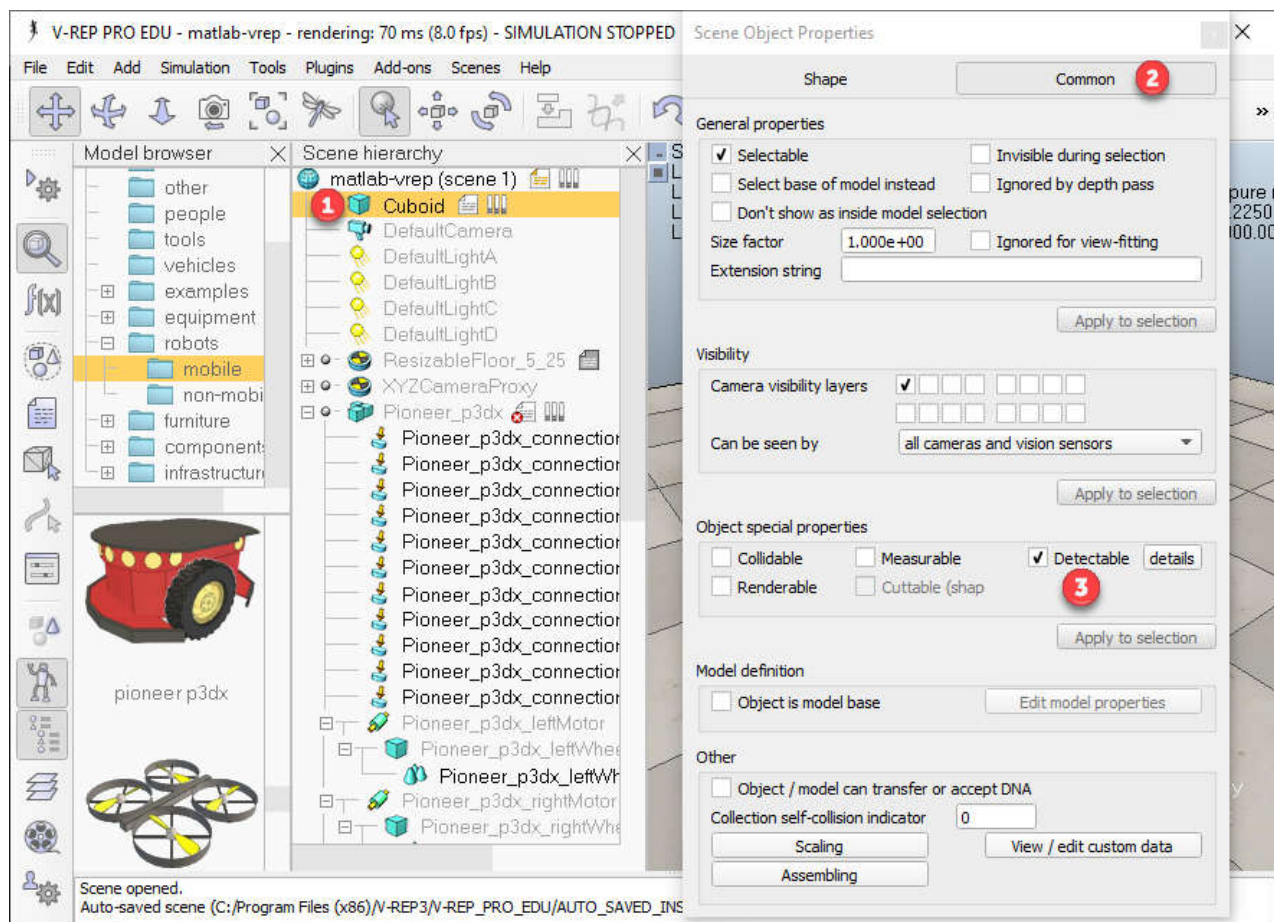
vrep.simxFinish(-1);
end
vrep.delete();

```

⇒ Simuler sur Matlab et V-REP le code et conclure.

2- Ajout d'un capteur ultrason :

A V-REP, dans la « Scene hierarchy », cliquer deux fois sur « cuboid ». Dans la fenêtre qui s'affiche cliquer sur « Common » puis cocher « Detectable »



Dans Matlab, écrire le code suivant :


```

vrep=remApi('remoteApi');
vrep.simxFinish(-1);
clientID=vrep.simxStart('127.0.0.1',19999,true,true,5000,5);

if (clientID>-1)
    disp('Connected')
    %%%

[returnCode,left_motor]=vrep.simxGetObjectHandle(clientID,'Pioneer_p3dx_leftMotor',vrep.simx_opmode_blocking);

[returnCode,front_Sensor]=vrep.simxGetObjectHandle(clientID,'Pioneer_p3dx_ultrasonicSensor5',vrep.simx_opmode_blocking);
    %%%

[returnCode]=vrep.simxSetJointTargetVelocity(clientID,left_motor,3,vrep.simx_opmode_blocking);

[returnCode,detectionState,detectedPoint,~,~]=vrep.simxReadProximitySensor(clientID,front_Sensor,vrep.simx_opmode_streaming);

for i=1:50

[returnCode,detectionState,detectedPoint,~,~]=vrep.simxReadProximitySensor(clientID,front_Sensor,vrep.simx_opmode_buffer);
disp(norm(detectedPoint));
pause(0.1);

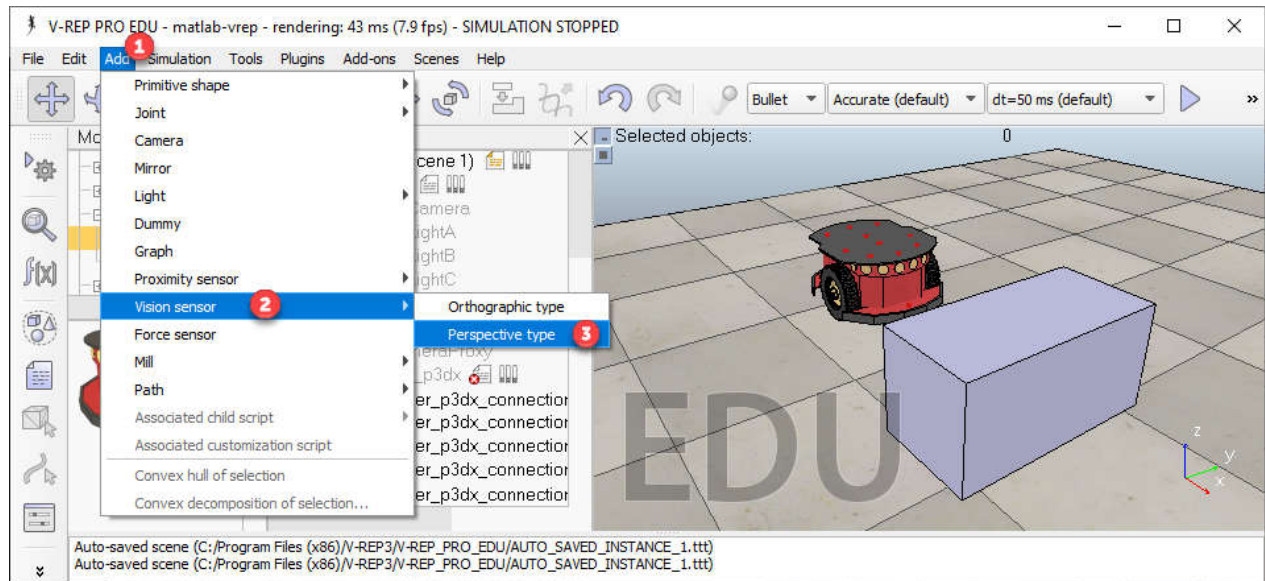
end

[returnCode]=vrep.simxSetJointTargetVelocity(clientID,left_motor,0,vrep.simx_opmode_blocking);
vrep.simxFinish(-1);
end
vrep.delete();

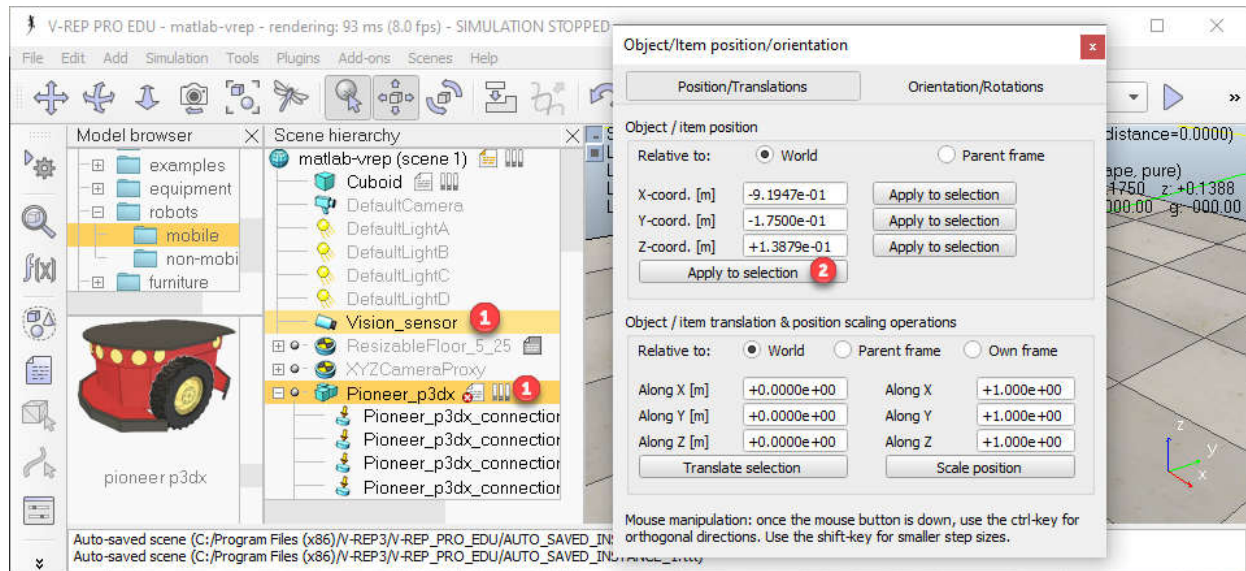
```

⇒ Simuler sur Matlab et V-REP le code et conclure.

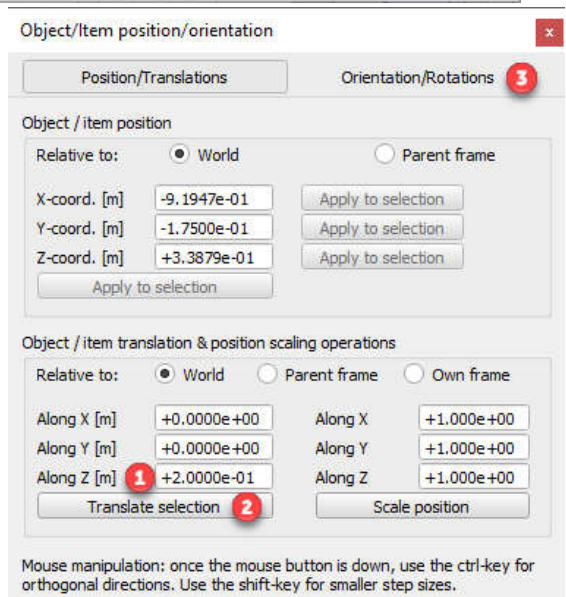
3- Ajout d'une caméra :



A la « Scene hierarchy », Selectionner en même temps « Vision_Sensor » et « Pioneer_p3dx », Dans la fenêtre qui s'affiche cliquer sur « Apply to selection »

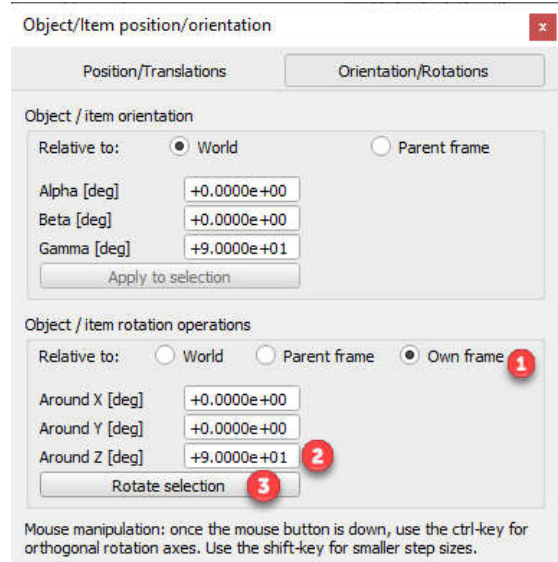


A la « Scene hierarchy », Selectionner seulement « Vision_Sensor » et la déplacer selon l'axe Z « Along Z [m] » de valeur 0.2m et valider par l'appuie sur « translate selection », Cliquer ensuite sur « Orientation/Rotations »



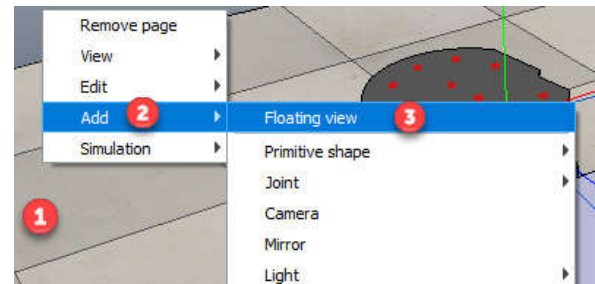
Dans « Orientation/Rotations », cliquer sur « Own frame » mettre la valeur d'orientation « Around Y [deg] » de la camera selon l'axe Y à 90° puis clique une seule fois sur « Rotate selection »

Mettre la valeur « Around Y [deg] » à zéro et mettre la valeur d'orientation « Around Z [deg] » de la camera selon l'axe Z à 90° puis clique une seule fois sur « Rotate selection »

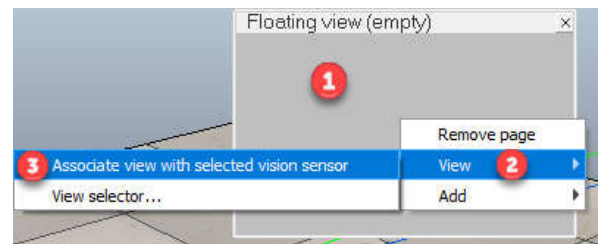


Pour voir la vue de la camera procéder comme suit :

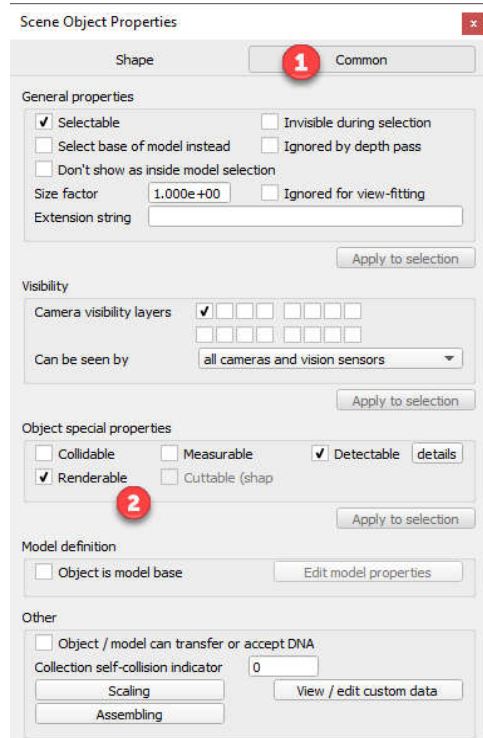
Cliquer par le bouton droit de la souris sur la scène et sélectionner « Add », « Floating view ».



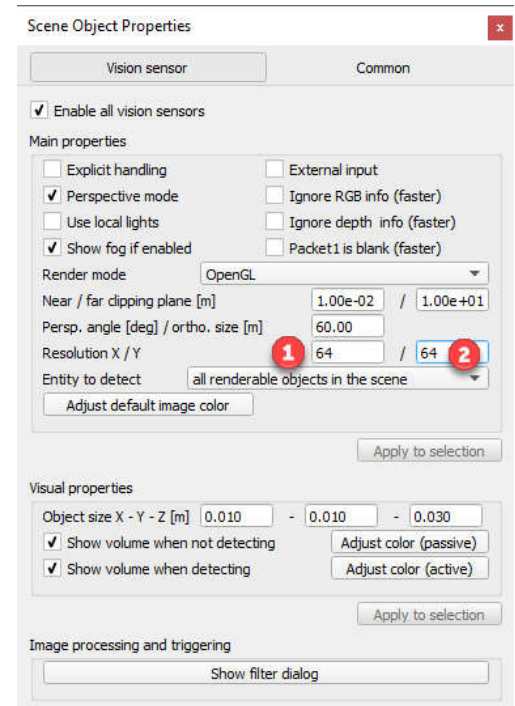
Cliquer par le bouton droit sur « Floating view », choisir « view » puis « Associate view with selected vision sensor »



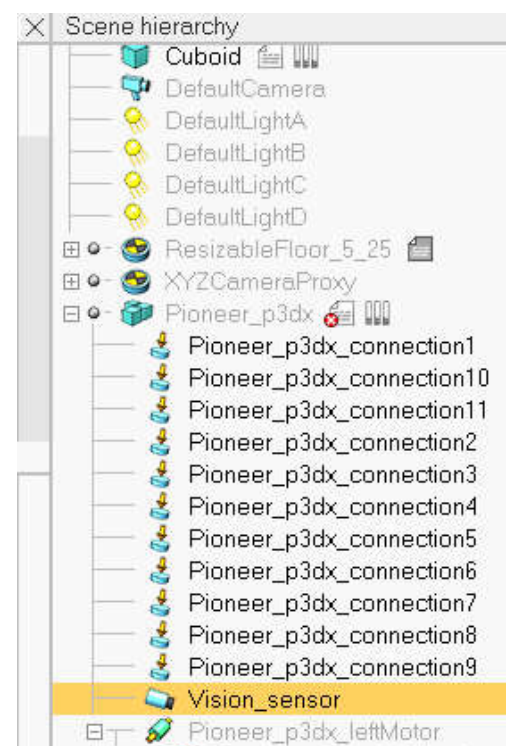
Pour que le camera peut voir « Cuboid » aller à la « Scene hierarchy », cliquer deux fois sur « Cuboid » une fenêtre s'affiche, choisir « Common » puis cocher « Renderable »



On peut augmenter la résolution du camera. Dans la « Scene hierarchy », cliquer deux fois sur « Vision_sensor » dans la fenetre qui s'affiche modifier la « resolution X/Y » à 256.



Mettre « Vision_sensor » dans « Pioneer_p3dx » pour qu'il fait partie du robot.



⇒ Simuler sur Matlab et V-REP le code suivant et conclure.

```
vrep=remApi('remoteApi');
vrep.simxFinish(-1);
clientID=vrep.simxStart('127.0.0.1',19999,true,true,5000,5);

if (clientID>-1)
    disp('Connected')
    %%%

[returnCode,left_motor]=vrep.simxGetObjectHandle(clientID,'Pioneer_p3dx_leftMotor',vrep
.simx_opmode_blocking);

[returnCode,front_Sensor]=vrep.simxGetObjectHandle(clientID,'Pioneer_p3dx_ultrasonicSen
sor5',vrep.simx_opmode_blocking);

[returnCode,camera]=vrep.simxGetObjectHandle(clientID,'Vision_sensor',vrep.simx_opmode_
blocking);
    %%%

[returnCode]=vrep.simxSetJointTargetVelocity(clientID,left_motor,3,vrep.simx_opmode_blo
cking);

[returnCode,detectionState,detectedPoint,~,~]=vrep.simxReadProximitySensor(clientID,fro
nt_Sensor,vrep.simx_opmode_streaming);

[returnCode,resolution,image]=vrep.simxGetVisionSensorImage2(clientID,camera,0,vrep.sim
x_opmode_streaming);

    for i=1:50

[returnCode,detectionState,detectedPoint,~,~]=vrep.simxReadProximitySensor(clientID,fro
nt_Sensor,vrep.simx_opmode_buffer);

[returnCode,resolution,image]=vrep.simxGetVisionSensorImage2(clientID,camera,0,vrep.sim
x_opmode_buffer);
        imshow(image)
        disp(norm(detectedPoint));
        pause(0.1);
    end

[returnCode]=vrep.simxSetJointTargetVelocity(clientID,left_motor,0,vrep.simx_opmode_blo
cking);
vrep.simxFinish(-1);
end
vrep.delete();
```

