

# Création d'une base mobile avec bras articulé pour la saisie et la manipulation d'objets en RV et implémentation de techniques d'Interaction 3D adaptées.

Projet RV, MASTER 1, EFREI, PR n° 1 du 16 octobre 2023.

Frédéric Davesne

## 0. Introduction

Il s'agit de reprendre le cahier des charges construit au TD n°2 (voir exemple de corrigé sur <https://ibisc.univ-evry.fr/~fdavesne/ens/>). Pour cela, nous allons mettre en oeuvre les différentes étapes de construction de l'application qui y sont données:

- variables d'état et scripts associés à leur gestion ;
- machines à états ;
- graphes de scène.

## 1. Étapes du PR n° 1

- a- Mise en place des articulations du bras articulé
  - Spécifier la hiérarchie des points pivots des axes de rotation du bras articulé ;
  - Inclure les différents objets graphiques composant le bras articulé dans la hiérarchie des points pivots ;
- b- Mise en place des éléments graphiques
  - Éléments de la base mobile et du bras articulé
  - Objets à saisir par le robot mobile
- c- *Premiers scripts*
  - Construire les scripts associés à la gestion de l'interactivité de la base mobile et du bras robotique: techniques d'I3D utilisées, machines à états tels que spécifiées dans le cahier des charges (voir le TD n° 2)

## 2. Pré-requis et assistance

### Pré-requis

- Le cours de Réalité Virtuelle et, en particulier, les techniques d'interaction 3D ;
- La correction du TD n° 2 présente sur le WEB (<https://ibisc.univ-evry.fr/~fdavesne/ens/>).

### Assistance

- Concernant le cours d'Interaction Homme/Machine, voir sur <https://ibisc.univ-evry.fr/~fdavesne/ens/>
- Concernant le projet en lui-même, les éléments concernant le robot (objets graphiques, script de gestion du bras articulé *gere\_bras.cs* (modèle inverse, documentation) se récupèrent ici: [https://ibisc.univ-evry.fr/~fdavesne/ens/ens\\_efrei\\_m1/proj/data/elements\\_robot.zip](https://ibisc.univ-evry.fr/~fdavesne/ens/ens_efrei_m1/proj/data/elements_robot.zip)

## 3. Mode opératoire

### 3.0. Fonctionnement

Vous serez amené à chercher en groupe vos solutions aux problèmes posés par *Unity*. Les aides en ligne sont suffisamment bien construites pour que vous puissiez le faire ... Alors, à votre Google préféré !

### 3.1. A la fin de la séance ...

Remettez-moi l'état de votre projet *Unity*. **Pour cela, veuillez suivre la procédure donnée dans l'étape 6 du PR n°1.**

### 3.2. Construction du robot mobile

Objectif: Cette section va aboutir à la création de votre première scène sous *Unity* que vous appellerez *Scène 1*. Elle contiendra le robot mobile avec des points pivots correctement placés qui permettront d'exécuter les techniques d'I3D de navigation, sélection et manipulation sur le robot mobile.

Etape 1: Création d'un projet *Unity* nommé **Projet\_robot**.

Etape 2: Création d'un répertoire **robot mobile** dans la fenêtre **Project** de *Unity*, puis d'un répertoire **graphique** sous **robot mobile**;

Etape 3: Importation des objets du bras articulé et de la base mobile à partir des éléments graphiques contenus dans le fichier *elements\_robot.zip*, **répertoire Graphique/robot\_complet/FBX** et placement dans la hiérarchie du projet **robot\_mobile/graphique par un drag&drop de l'ensemble des éléments de robot/FBX**;

Etape 4: Création d'un *empty object* nommé *robot* dans la fenêtre **Hierarchy** de *Unity* (*GameObject* -> *Create Empty*). En cliquant sur *robot*, bien vérifier dans la fenêtre **Inspector** onglet **Transform** que la position de robot est (0,0,0) ainsi que son orientation. Sinon fixer le tout à 0. Puis drag&drop des objets graphiques de la fenêtre **Project** de *Unity* sous *robot* dans la fenêtre **Hierarchy**.

Etape 5: Rotation de 180° selon l'axe des y de *robot*. Résultat: les objets fils de *robot* seront pivotés de 180° selon l'axe y.

Etape 6: Sauvegarde de la scène en « **PR 1 - Scène 1 – Etape 6** » (Save Scene As, Scène 1) dans le répertoire *Scenes* sous *Assets*.

**Créer un package *projet\_robot* de sauvegarde de votre travail. Pour cela, aller dans Assets -> Export package et cocher tous les items le cas échéant.**

**Si besoin, vous pourrez reprendre votre travail en important ce package à partir d'un projet *Unity* vierge (Assets -> Import package -> Custom package)**

Etape 7: Construction des points pivots du robot. Pour cela, lire les documents **Spécification des points pivots du bras articulé FANUC**:

- A chaque création de points pivots  $O_k$ , bien vérifier que celui-ci est bien inscrit graphiquement dans le bras articulé du robot en cliquant sur son *Empty object* dans la fenêtre **Hierarchy** et en regardant le repère

(mode *Pivot/Local*) associé à l'objet dans la fenêtre *Scene* de *Unity*. Faire bien attention aux valeurs en Position et Rotation. Nous rappelons que les *Oktierce* doivent avoir *Position* (0,0,0) et *Rotation* (0,0,0) ;

- Dans fenêtre ***Hierarchy***, faites les drag&drop suivants de l'Empty object *robot* vers la hiérarchie des points pivots du robot après la création des *Empty object Ok, Okprime et Oktierce*:

- base\_robot, roue gauche, roue droite, membre1 sous O0 ;
- membre2 sous O1tierce ;
- membre31 et membre32 sous O2tierce ;
- membre4 sous O3tierce ;
- membre5 sous O4tierce ;
- membre6 sous O5tierce ;
- membre7 et membre8 sous O6tierce ;
- pince\_g sous Opince\_g.

Etape 8: Création de la pince droite du bras articulé

- cliquer sur pince\_g dans la fenêtre *Hierarchy* puis faire un copier/coller. Renommer le nouvel élément par pince\_d et faire un drag&drop vers Opince\_d ;
- modifier la Partie Position de Opince\_d en (0,017,0,0) et vérifier que l'apparence de la pince générale est correcte.

Vous pouvez à présent supprimer l'*Empty Object robot*.

Le graphe de scène dans la fenêtre *Hierarchy* doit ressembler à [cela](#).

Etape 9: Vérification de la cohérence les points pivots et du modèle graphique du robot

- Effectuer une rotation de 20° selon l'axe Y de O1tierce Rotation (0,20,0), observer le résultat et remettre O1tierce à Rotation (0,0,0). Faire de même avec O3tierce et O5tierce.
- Vérifier que *Otool* semble être bien placé à chaque essaie de rotation.

Sauvegarder votre scène *Unity* que vous nommerez « **PR 1 - Scène 2 – Etape 9** » dans le répertoire scènes.

### **3.3. Test de l'interactivité du bras articulé**

Objectif: Le script C# gérant les déplacements articulaires du bras du robot s'appelle *gere\_bras.cs* et est déjà écrit. Il s'agit de le mettre en œuvre dans *Unity* et de le tester afin de vérifier la validité de ce qui a été réalisé dans 3.2 .

Etape 10: Création d'un répertoire ***scripts*** sous **robot mobile** dans la fenêtre **Project**. Placer le script *gere\_bras.cs* dans ce répertoire.

Etape 11: Associer le script *gere\_bras.cs* à l'Empty Object *O0*.

- Pour cela, faire un drag&drop de *gere\_bras.cs* à partir de la fenêtre *Project* vers O0 dans la fenêtre *Hierarchy*.

- Cliquer sur O0 dans la fenêtre *Hierarchy* et observer sa composante script dans la fenêtre *Inspector*.

- *gere\_bras.cs* possède 11 variables directement modifiables à partir de l'interface de *Unity*: O1tierce, O2tierce, O3tierce, O4tierce, O5tierce, O6tierce, Opince, Increment, Ok, num\_axe\_butee, DEBUG.

- Ouvrir *gere\_bras.cs* à partir de la fenêtre *Project* afin d'y voir le code. Constaté que ces variables sont des variables publiques de la classe *gere\_bras*.

- Associer O1tierce, O2tierce, O3tierce, O4tierce, O5tierce, O6tierce et Otool de la fenêtre *Hierarchy* aux paramètres O1tierce, O2tierce, O3tierce, O4tierce, O5tierce, O6tierce et Opince de la fenêtre *Inspector* pour O0 en effectuant un drag&drop de ceux-ci. On doit obtenir [ceci](#).

Etape 12: Exécuter le programme en cliquant sur Play. Que constatez-vous?

Etape 13: Vérification à partir de la Console de Unity.

- Afficher la fenêtre *Console* de Unity. Les coordonnées de *Opince* doivent être affichées et donner [0.44; 0.365; 0] dans le repère lié à O0 et dans le repère du monde.
- Cliquer sur O0 dans la fenêtre *Hierarchy* et taper 0.01 pour X de la variable *Increment* du script *gere\_bras* dans la fenêtre *Inspector*. Que constatez-vous dans la console? Faire de même pour Y et Z de la variable *Increment*. On doit obtenir [ceci](#).
- Modifier la position et/ou l'orientation de O0 et refaire les tests précédents. Que constatez-vous?
- Regarder dans le code C# de *gere\_bras.cs* les deux fonctions *get\_Opince\_absolute\_position()* et *get\_Opince\_local\_position()*. Comment procède-t-on pour obtenir la position/l'orientation de la pince par rapport à O0?

Sauvegarder votre scène Unity que vous nommerez « **PR 1 - Scène 3 – Etape 13** » dans le répertoire scènes.

### 3.4. Construction d'un premier script

Objectif: Piloter le bras du robot à l'aide du clavier, piloter la base mobile du robot à l'aide du clavier.

Etape 14: Dans la fenêtre *Project*, créer un script *ClavierScript.cs* dans le répertoire **robot mobile/scripts** et ouvrir *ClavierScript.cs* : une classe *ClavierScript* est déclarée ainsi que les fonctions principales de Unity: *Init()* exécutée au lancement du programme et *Update()* exécutée à chaque frame. Associer dans un premier temps ce script à O0.

Etape 15: Il faut pouvoir accéder à la variable *increment* du script *gere\_bras*. On va pour cela procéder comme suit:

- déclarer une variable publique de type *gere\_bras* dans la classe *ClavierScript*: `public gere_bras gere_bras=null;`
- Cette variable, appelée ici *gere\_bras* est une entrée de la classe et on doit fournir dans la partie *Inspector* l'objet contenant le script *gere\_bras*, c'est-à-dire O0.

Etape 16: Dans la fonction *Update()*, écrire un petit code permettant de mettre à jour la variable *increment* suivant l'appuie sur une des six touches suivantes: Flèche Gauche, Flèche Droite, Flèche Haut, Flèche Bas, A et Z, produisant des incréments de  $\pm 0.01$  sur chacun des trois axes. On utilisera *Input.GetKeyUp("touche")*. Bien regarder l'utilisation de cette fonction sur le Web.

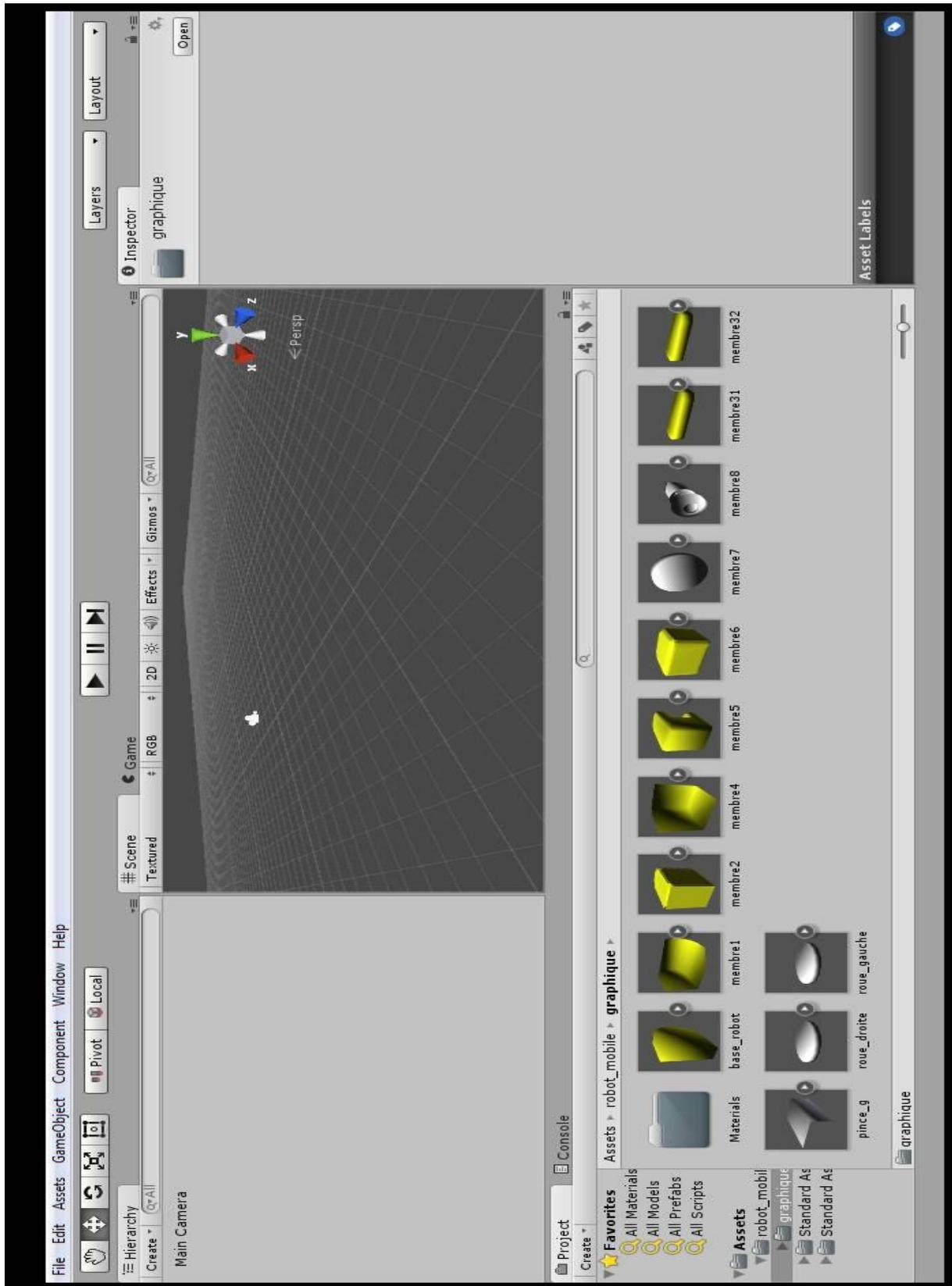
Etape 17: Tester le bon fonctionnement du script *ClavierScript* avec la console comme effectué dans 3.3.

Sauvegarder votre scène Unity que vous nommerez « **PR 1 - Scène 4 – Etape 17** » dans le répertoire scènes.

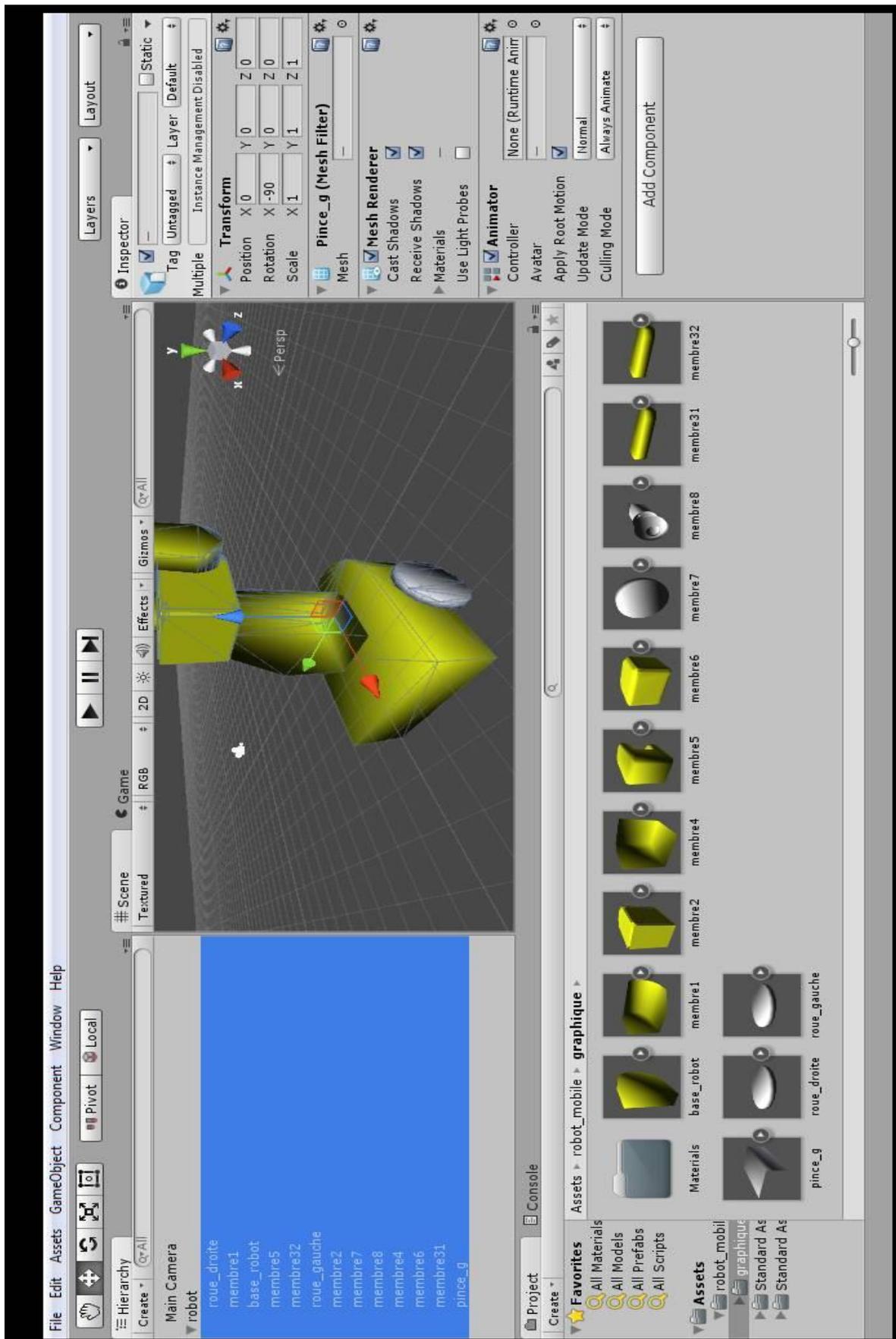
**FIN**

## **Annexes**

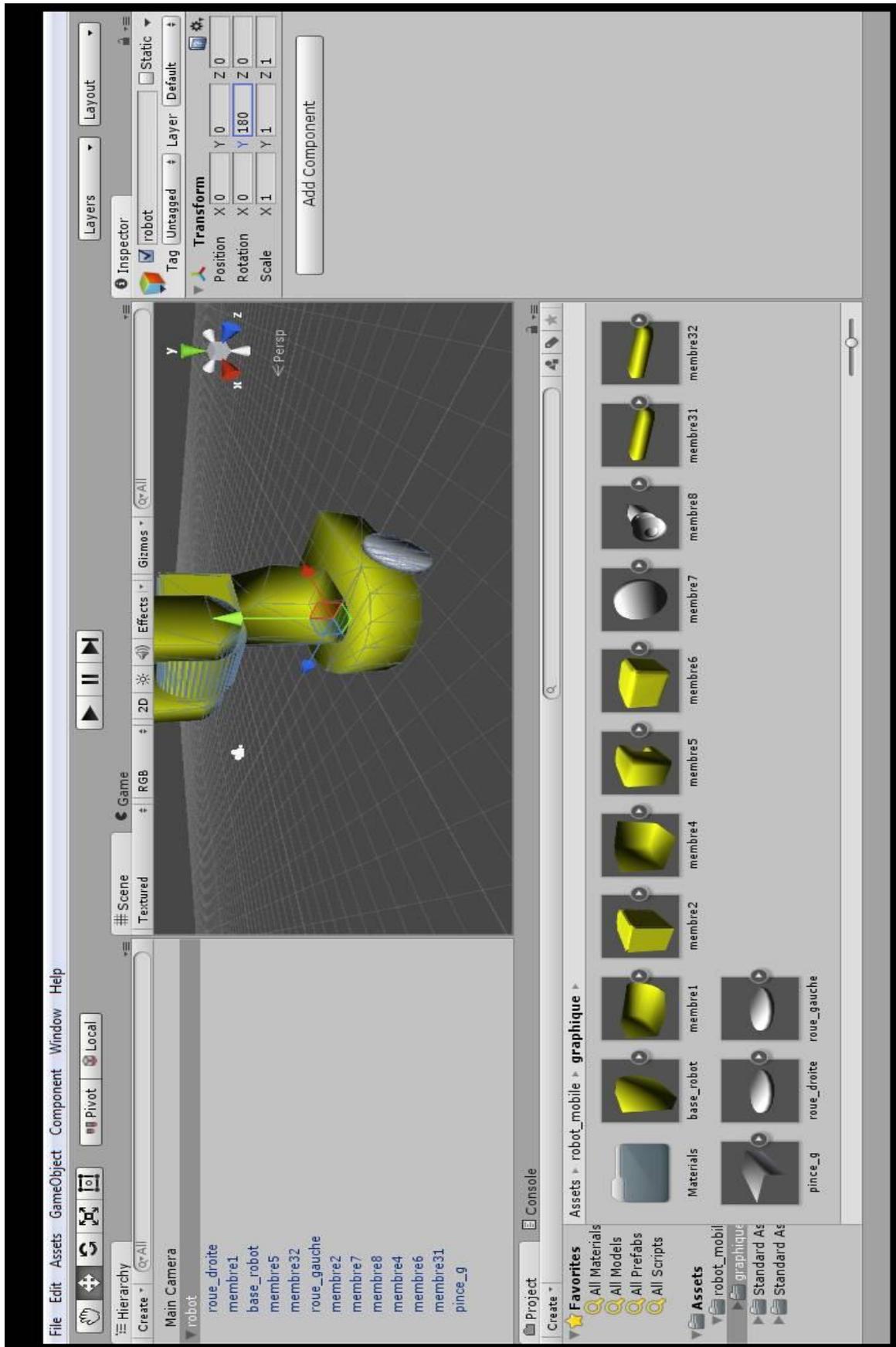
## Etapes 2 et 3:



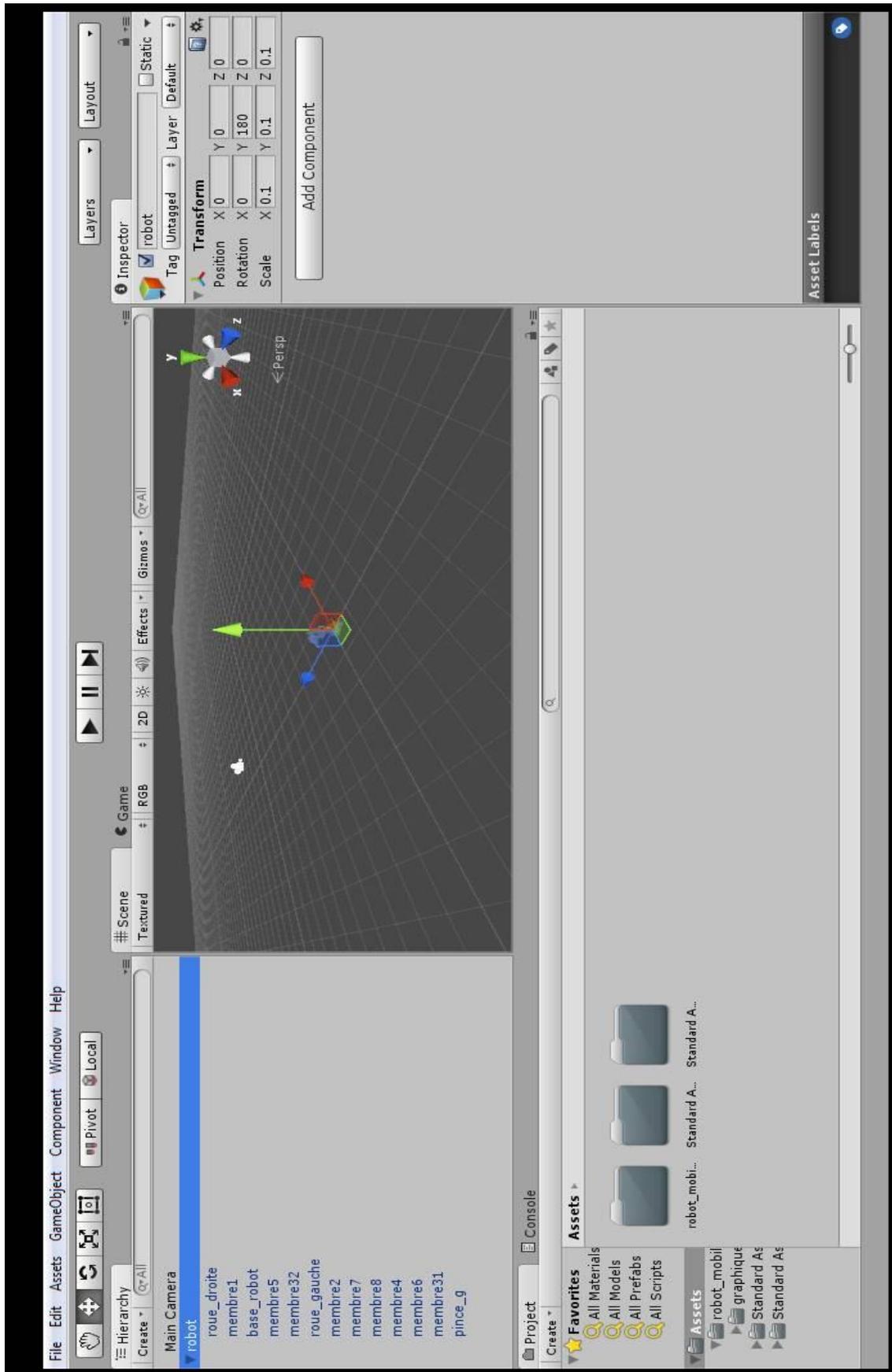
## Etape 4:



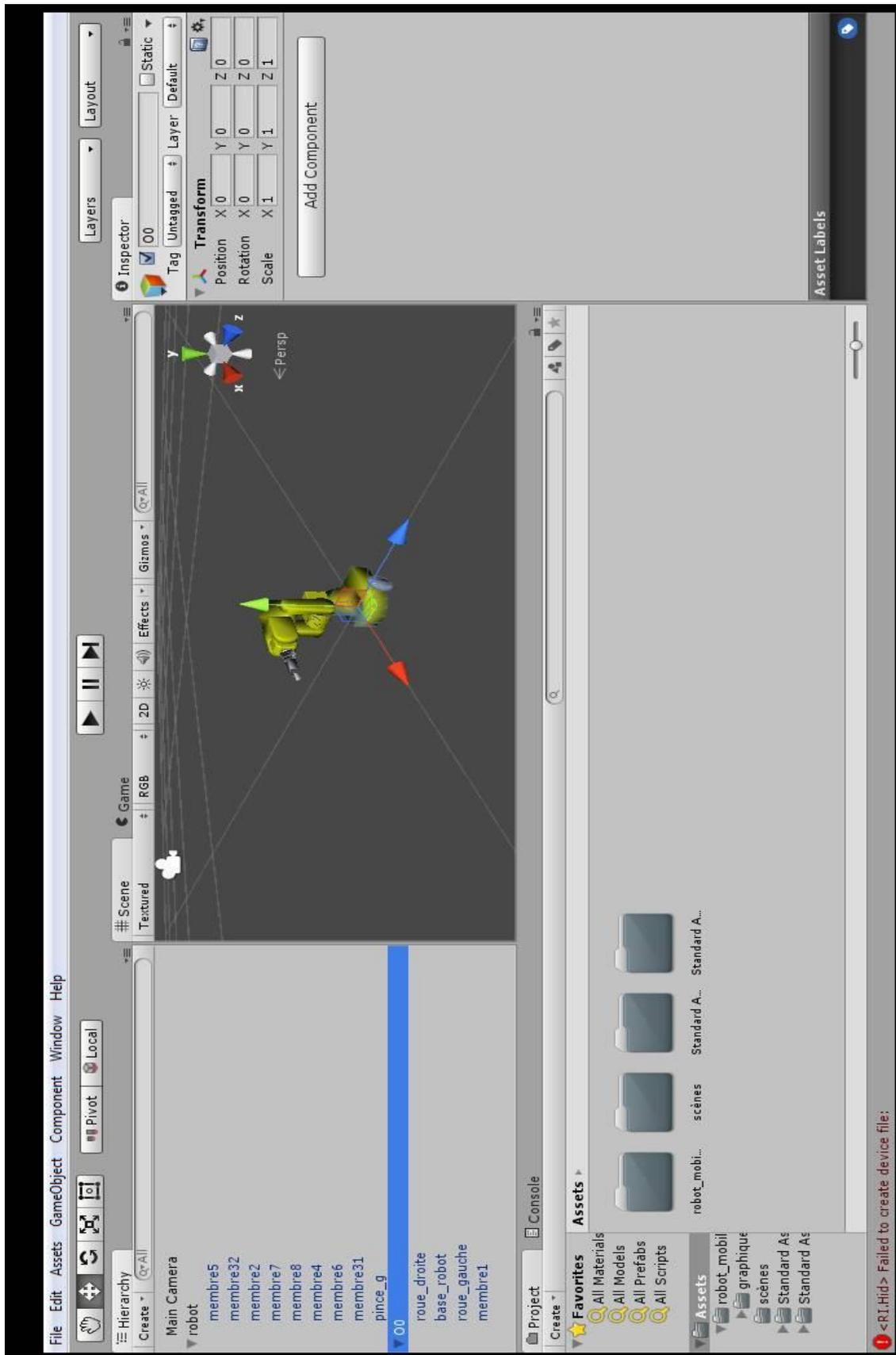
## Etape 5:



## Etape 6:

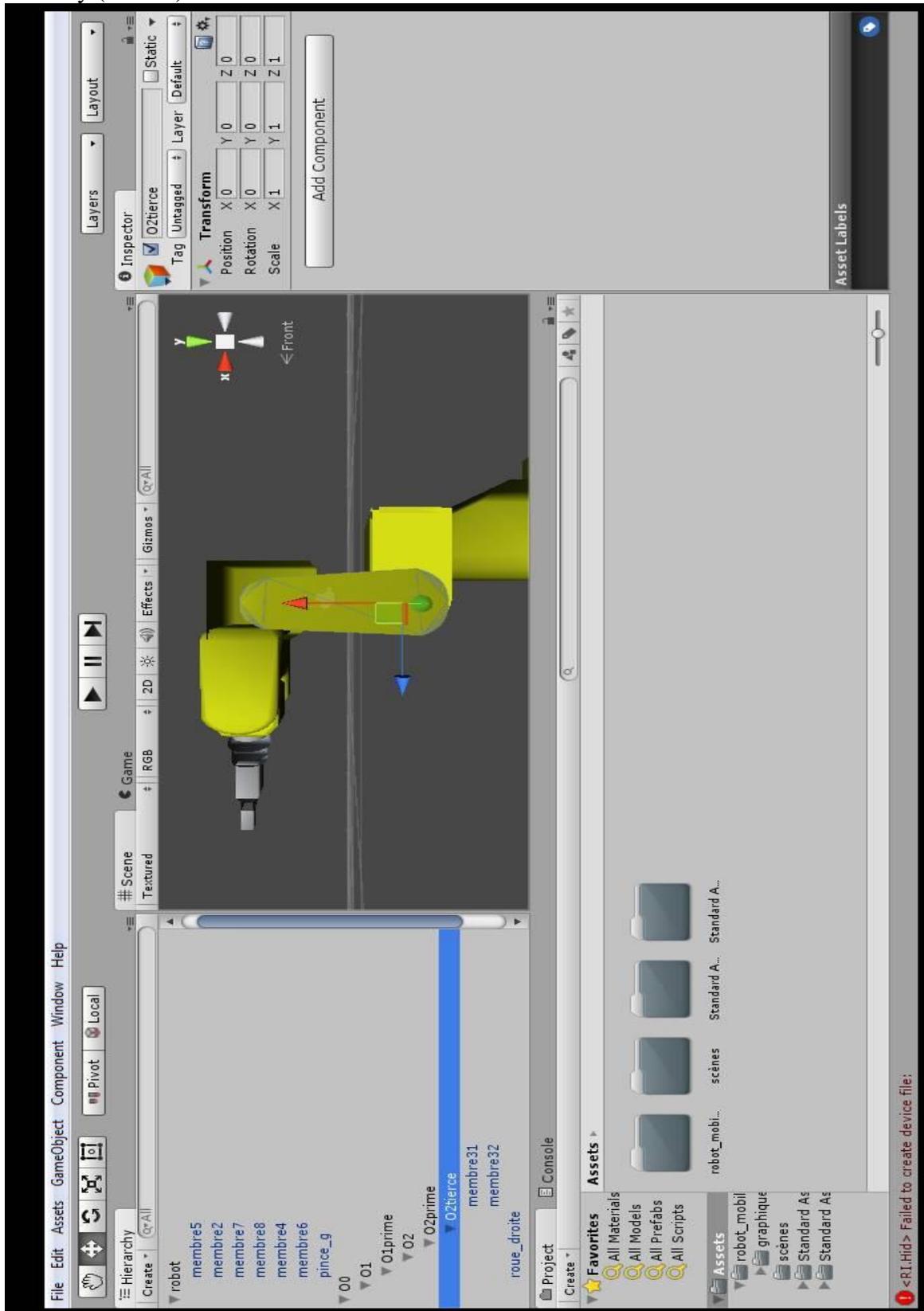


Etape 7: Création de O0 et drag&drop de roue\_gauche, roue\_droite, base\_mobile et membre1 sous O0.



**Etape 7: Création de O1, O1prime, O2, O2prime et O2tierce, puis drag&drop de membre31 et membre32 sous O2tierce.**

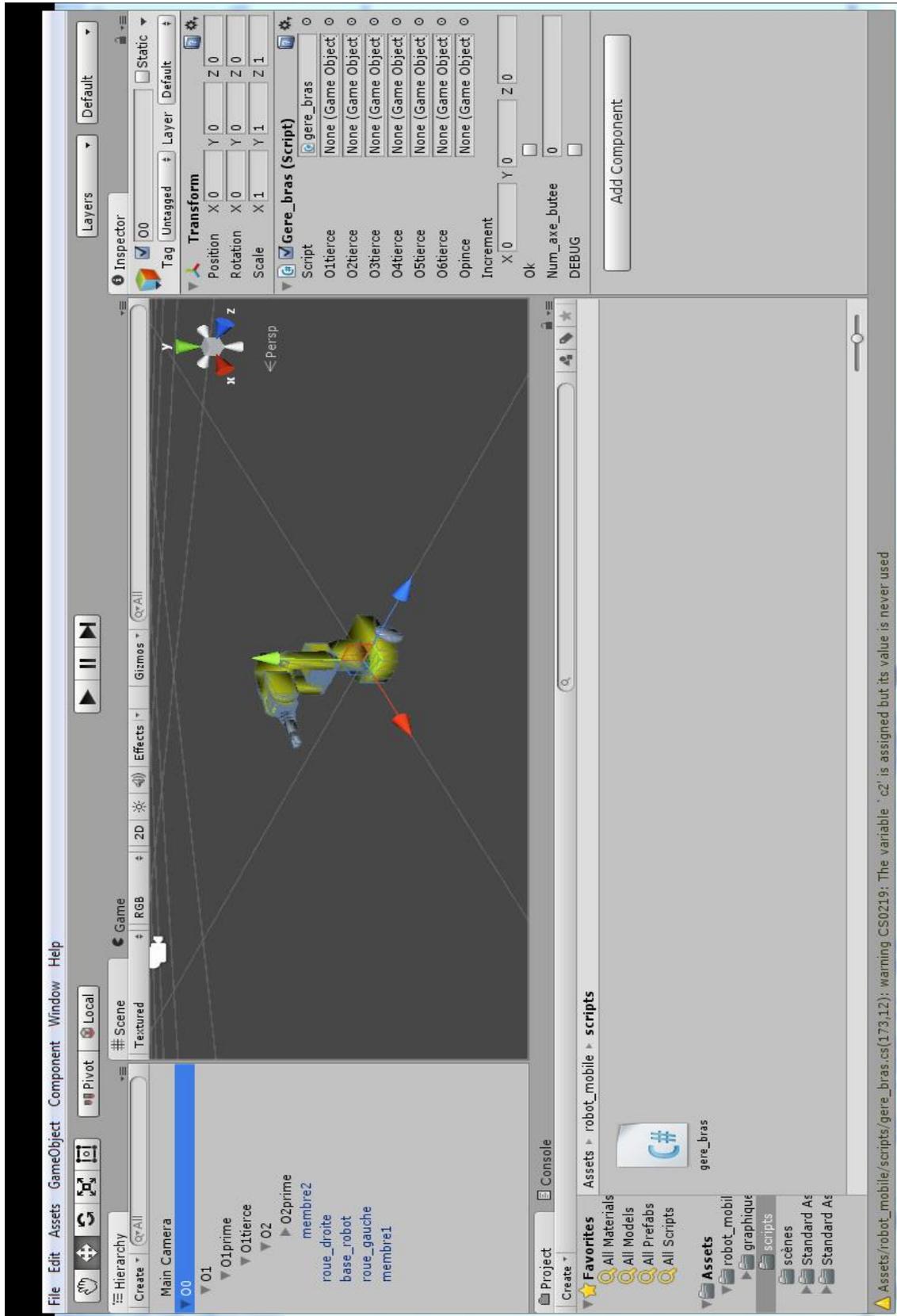
Le repère lié à O2tierce est apparent. L'axe de rotation de membre31 et membre32 est bien l'axe Oy (en vert).



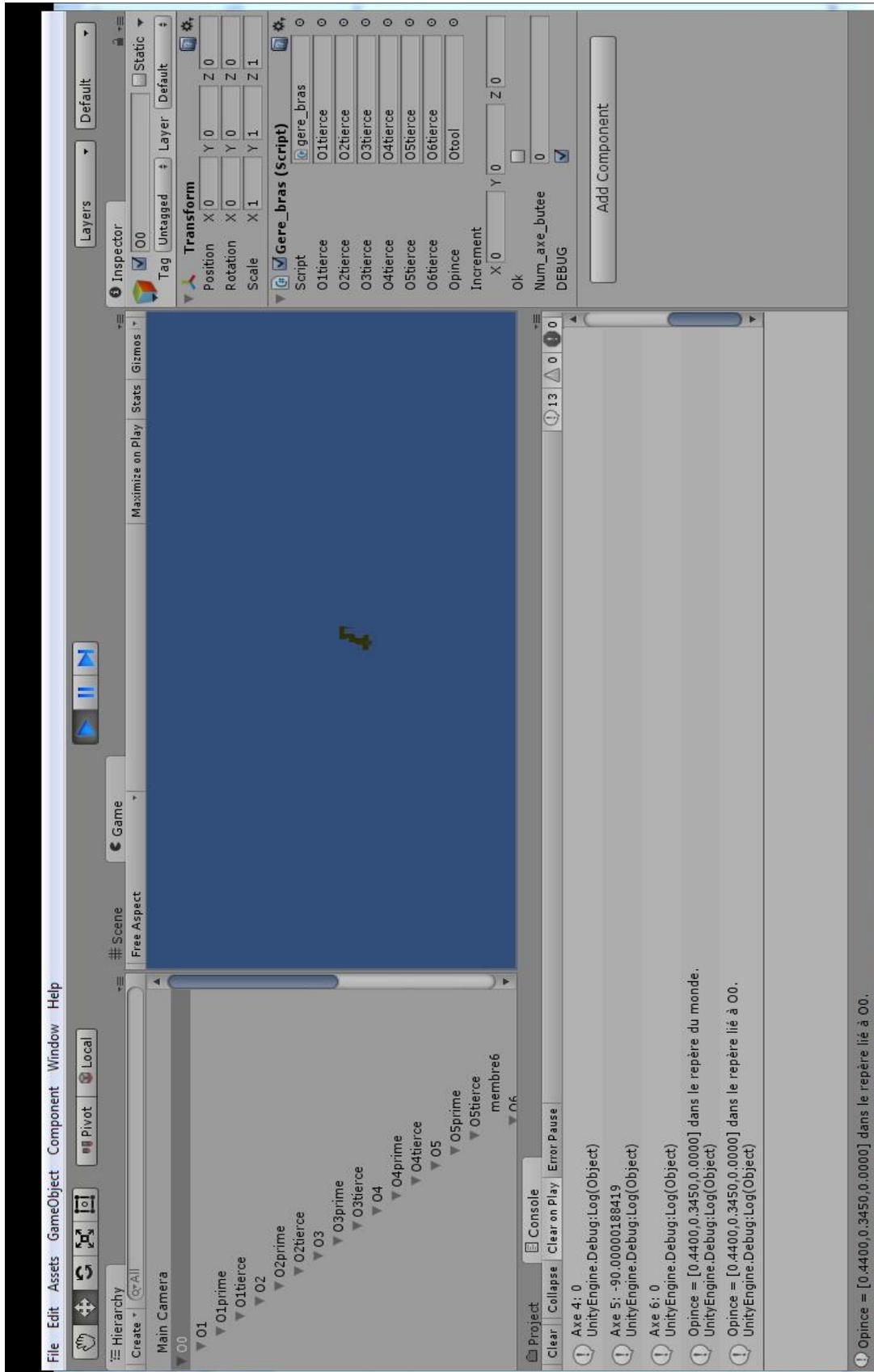


## Étape 11: Mise en place du script gere\_bras.cs sous O0 dans Hierarchy.

Il faut à présent remplir la partie paramètres d'entrée: 01tierce ...



## Etape 12: Exécution du programme et affichage de la position 3D de Otool dans la Console.



**Etape 17:**  
**Après 3 appuis sur la touche "Flèche Bas"**

