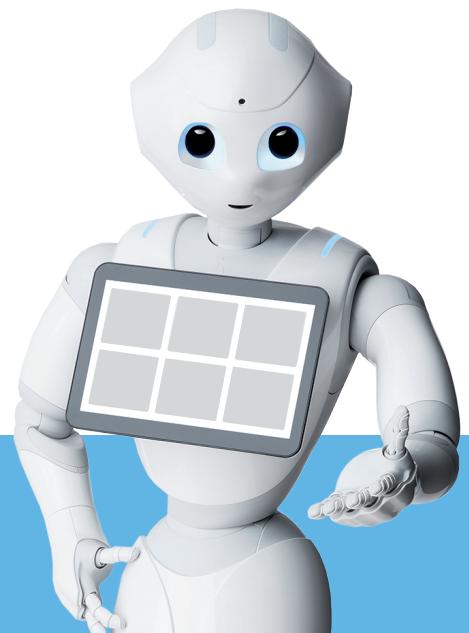
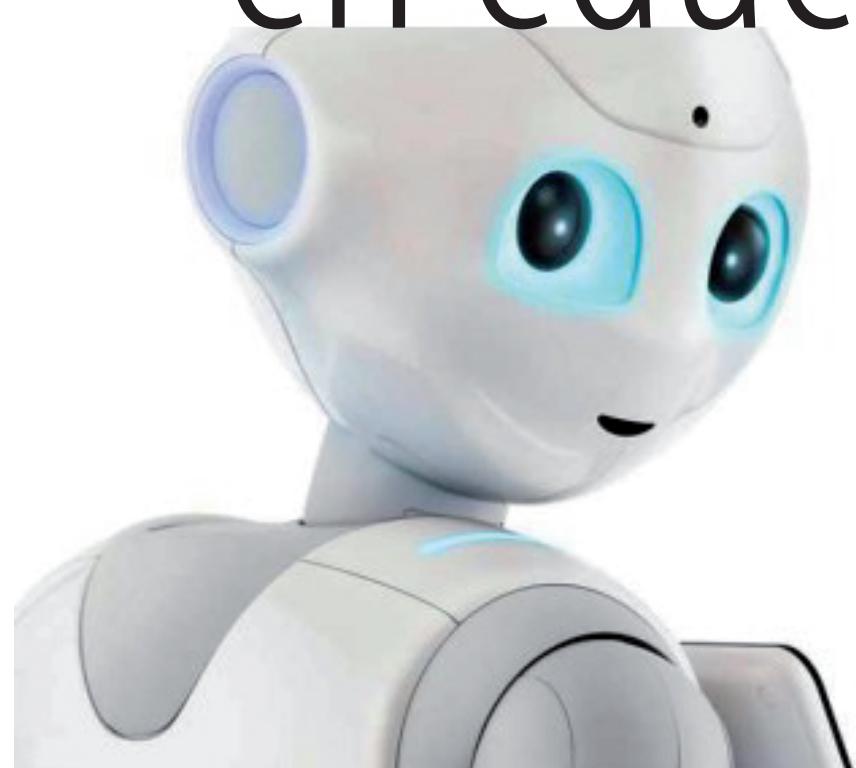


# Le robot PEPPER en éducation



## Guide d'utilisation

Thierry **KARSENTI**

Nicolas **KERBRAT**

JULIEN **BUGMANN**

SIMON **PARENT**

OCTOBRE 2018

## *Préface : PEPPER, le robot humanoïde*

PEPPER est un robot humanoïde, mesurant 120 cm et pesant 28 kilogrammes. Il a été développé par la société SoftBank Robotic pour communiquer avec les humains, grâce à sa voix et sa gestuelle, naturelles et intuitives.

PEPPER est capable d'entendre, de parler, d'identifier les principales émotions, de voir et de se déplacer. Toutes ces actions sont possibles grâce à ses caméras, ses microphones, ses haut-parleurs, ses capteurs sensoriels, sa tablette et ses trois roues multidirectionnelles.

PEPPER est principalement utilisé par des entreprises dans le but d'accueillir et d'informer les clients, ou par des familles japonaises comme compagnon au quotidien. Mais pourquoi ne pas l'utiliser, de la même manière que NAO, son précurseur, comme outil de programmation ?

# Table des matières

<i>Préface : PEPPER, le robot humanoïde.....</i>	2
<b>1. Découverte de PEPPER.....</b>	6
<b>1.1 Présentation du robot .....</b>	6
<b>1.2. Déplacements manuels de Pepper .....</b>	6
1.2.1 Lorsqu'il est éteint : .....	6
1.2.2 Lorsqu'il est allumé : .....	6
1.2.3 Utilisation des clés de déblocage des freins : .....	7
<b>1.3. Démarrage et arrêt de Pepper.....</b>	8
1.3.1 Démarrer Pepper : .....	8
1.3.2 Éteindre Pepper : .....	8
1.3.3 Éteindre Pepper avec l'arrêt forcé : .....	8
1.3.4 Charger Pepper : .....	9
1.3.5 Mettre en veille Pepper (mode repos) : .....	10
<b>1.4 Configuration de Pepper .....</b>	10
<b>1.5. Menu des paramètres .....</b>	16
1.5.1 La page Mon robot.....	16
1.5.2 La page Paramètres réseau .....	17
1.5.3 La page Mise à jour des applications.....	19
1.5.4 La page Paramètres avancés .....	21
<b>1.6. Vie autonome .....</b>	22
1.6.1 Parler avec le robot : .....	22
1.6.2 <i>Basic channel</i> : Que puis-je dire à Pepper ? .....	23
1.6.3 Autres fonctionnalités de Basic Channel : .....	27
D'autres réactions du mode autonome : .....	27
1.6.4 Désactiver la vie autonome : .....	27
1.6.5 Améliorer la vie autonome : .....	27
<b>1.7. Fermeture d'une activité .....</b>	28
<b>1.8. Les voyants d'état .....</b>	28
<b>1.9. Page web de Pepper .....</b>	28
1.9.1 Accéder à la page Web du robot sur un navigateur web : .....	28
1.9.2 Accéder à la page Web du robot sur Choregraphe : .....	29
<b>1.10. Téléchargement d'une application sur le Store Aldebaran.....</b>	30
1.10.1 Sur la tablette du robot : .....	30
1.10.2 Sur un ordinateur : .....	30
<b>2 : Utilisation du logiciel Choregraphe .....</b>	31
<b>2.1 Présentation du logiciel.....</b>	31
<b>2.2 Installation .....</b>	31

<b>2.3 Interface .....</b>	<b>31</b>
2.3.1. Les fenêtres affichées par défaut .....	32
2.3.2. Personnalisation de l'interface.....	37
2.3.3. Description de la barre d'outils :.....	39
<b>2.4 Connexion au robot .....</b>	<b>40</b>
2.4.1. Connexion WiFi.....	40
2.4.2. Utilisation d'un routeur WiFi multifonction .....	42
2.4.3. Connexion Filaire.....	43
2.4.4. Débogage de connexion .....	43
<b>2.5 Cration d'un nouveau projet.....</b>	<b>44</b>
<b>2.6. Les boes de comportements prexistantes .....</b>	<b>48</b>
2.6.1. Animation.....	48
2.6.2 Speech (Discours).....	51
2.6.3 LEDs (Diodes lectroluminescentes).....	54
2.6.4 Multimedia .....	57
2.6.5 Movement (Mouvements) .....	60
2.6.6 Sensing (Dtection).....	66
2.6.7 Programming (Programmation).....	75
<b>2.7. Generalites sur les entrees et les sorties .....</b>	<b>87</b>
<b>2.8. Cration de nouvelles boes .....</b>	<b>89</b>
2.8.1 Diagram.....	89
2.8.2 Timeline .....	91
2.8.3 Python .....	103
2.8.4 Dialog.....	110
<b>2.9 Cration d'un comportement autonome.....</b>	<b>117</b>
<b>3. Les fonctionnalites de PEPPER .....</b>	<b>123</b>
<b>3.1 Parler.....</b>	<b>123</b>
3.1.1 Boe Say.....	123
3.1.2 Boe Animated Say .....	124
<b>3.2 Se dplacer .....</b>	<b>125</b>
3.2.1 Move To .....	125
3.2.2 Move Along.....	128
<b>3.3 Utilisation de la tablette.....</b>	<b>132</b>
3.3.1 Afficher une image .....	132
3.3.2 Afficher une vido .....	133
3.3.3 Afficher un fichier html.....	135
3.3.4 Afficher un site internet.....	137
<b>3.4 Jouer de la musique .....</b>	<b>139</b>
<b>3.5 Poser des questions et valider les reponses .....</b>	<b>142</b>
<b>3.6 Reconnatre des personnes .....</b>	<b>145</b>

<b>3.7. Reconnaître des objets .....</b>	<b>149</b>
3.7.1 Suivre la balle rouge .....	149
3.7.2 Apprendre à reconnaître des objets.....	152
3.7.3 Utiliser les NAOmarks.....	155
<b>4. Exemples de créations d'applications.....</b>	<b>157</b>
<b>4.1. L'histoire du Petit chaperon rouge .....</b>	<b>157</b>
<b>4.2. Jeu de quilles.....</b>	<b>166</b>
<b>4.3. Jeu des capteurs.....</b>	<b>177</b>

# **1. Découverte de PEPPER**

## **1.1 Présentation du robot**

PEPPER est un chouette compagnon mais également un excellent outil de programmation. L'autonomie de sa batterie est à peu près de 4 heures d'utilisation. Il peut être utilisé lorsqu'il est branché au secteur. Il a besoin de place pour se déplacer et écarter ses bras. Assure-toi qu'il soit sur une surface plane, pour qu'il roule en toute sécurité. Une chute pourrait être fatale. Il est capable de détecter les obstacles mais il ne peut pas les franchir.

Il est composé de 2 haut-parleurs situés au niveau des oreilles, de 4 microphones placés sur le dessus de sa tête et de 4 caméras, une sur le front, une dans la bouche et une dans chaque œil. Les LEDs au niveau des yeux, des oreilles et des épaules affichent des couleurs qui ont différentes significations.

9 capteurs tactiles sont placés un peu partout sur son corps (tête, tablette, mains et jambe) et sont utilisés pour diverses applications ou pour la programmation.

## **1.2. Déplacements manuels de Pepper**

### **1.2.1 Lorsqu'il est éteint :**

1. Placez une main à l'avant d'une de ses épaules et une main sur ses hanches. Cela lui évitera de basculer.
2. Poussez-le avec délicatesse. Soyez prudent aux obstacles.

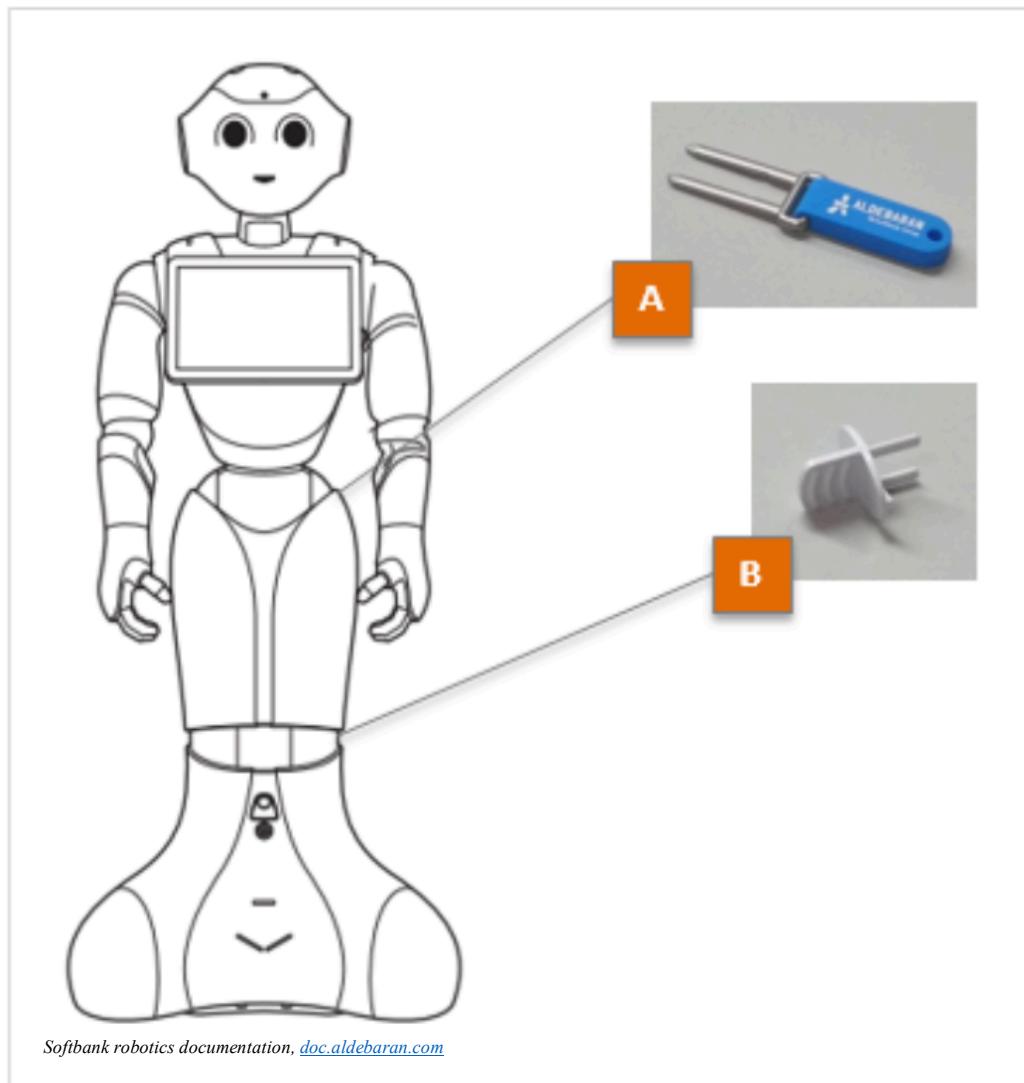
### **1.2.2 Lorsqu'il est allumé :**

1. Placez le robot en mode repos en plaçant votre main 3 secondes sur sa tête : paume sur la caméra du front et doigt sur les capteurs tactiles. Les yeux deviennent violets et le robot émet un son lorsqu'il s'endort.
2. Poussez-le de la même manière que lorsqu'il est éteint.

### 1.2.3 Utilisation des clés de déblocage des freins :

Pepper est composé de freins qui limitent la flexibilité de la jambe de Pepper. Avant de l'allonger, de le porter ou de l'emballer dans une boîte de rangement, il est fortement conseillé de déverrouiller ces freins. Pour cela, vous devez utiliser les deux clés de déblocage. Attention, vous devez les utiliser seulement lorsque Pepper est éteint. Lorsque vous les enlèvez, pensez à les ranger dans la partie molle du dos du robot.

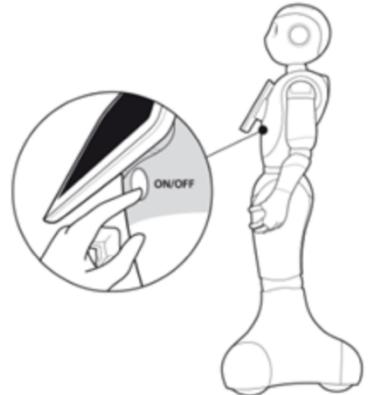
**NB :** Il est plus prudent d'être deux personnes pour porter Pepper et le mettre dans sa boîte de rangement. Idem, pour le sortir.



## 1.3. Démarrage et arrêt de Pepper

### 1.3.1 Démarrer Pepper :

1. Placez le robot dans un espace dégagé et sur une surface plane.
2. Assurez-vous que le robot soit chargé.
3. Assurez-vous que le bouton d'arrêt d'urgence soit déverrouillé.
4. Appuie une fois sur le bouton on/off situé sur la poitrine. La tablette s'allume et le robot se réveille. Tu aperçois des LEDS bleues sur les épaules du robot et tu entends un petit son. Cette étape peut prendre de plusieurs secondes à quelques minutes en fonction d'une éventuelle mise à jour.



Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)

Lorsque les voyants des épaules clignotent bleu, c'est que le robot effectue une mise à jour.

Lorsque les voyants des épaules clignotent rouge, c'est que le processus de démarrage a rencontré une erreur. Dans ce cas, redémarrer le robot.

### 1.3.2 Éteindre Pepper :

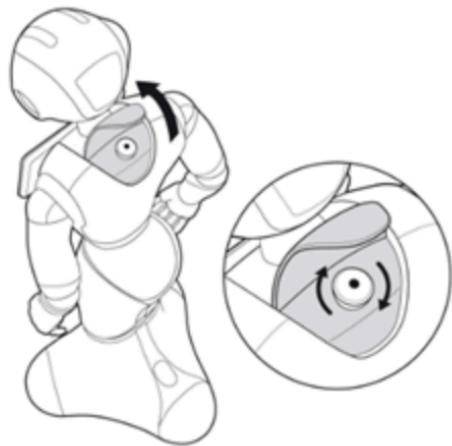
1. Assurez-vous que le robot soit sur une surface plate.
2. Restez appuyer sur le bouton on/off placé sur la poitrine du robot jusqu'à ce qu'il émette le son GNUK-GNUK. Le robot est arrêté lorsque tous les voyants sont éteints. Cette étape prend environ 20 secondes.
3. Déplacez Pepper manuellement jusqu'à un endroit sécuritaire ou mets le dans sa boîte de rangement.

### 1.3.3 Éteindre Pepper avec l'arrêt forcé :

Il est possible que vous rencontriez des difficultés pour éteindre Pepper ou que vous rencontriez des bogues. Vous pouvez également avoir besoin d'effectuer un arrêt d'urgence. Dans ces cas-là, vous devez rester appuyer 8 secondes sur le bouton on/off ou alors vous servir du bouton d'arrêt d'urgence situé dans le haut du dos du robot.

1. Tirez la partie molle derrière les épaules.
2. Tournez le bouton dans le sens des aiguilles d'une montre. Il doit remonter.
3. Replacez la partie molle.

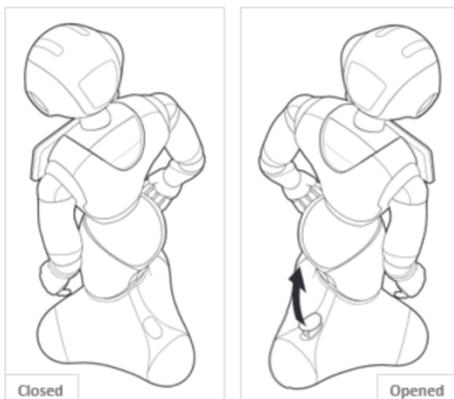
Avant de redémarrer le robot, pensez à déverrouiller le bouton d'arrêt d'urgence.



*Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)*

#### 1.3.4 Charger Pepper :

1. Branchez le chargeur sur le secteur. Un voyant rouge s'affiche sur le chargeur.
2. Ouvrez la trappe d'alimentation.
3. Branchez le chargeur au robot. Pour cela, insérez la prise correctement et tournez la dans le sens des aiguilles d'une montre jusqu'à ce qu'elle s'enclenche.
4. Le voyant du chargeur devient vert lorsque le robot est chargé à 100%. Les DELs des épaules clignotent vert si le robot est allumé.



*Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)*



*Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)*

**NB :** La trappe d'alimentation doit être fermée pour que Pepper puisse se déplacer. Il ne peut donc pas le faire lorsqu'il est branché au secteur.

### 1.3.5 Mettre en veille Pepper (mode repos) :

Il peut être utile de mettre en veille Pepper lorsque tu veux le réutiliser peu de temps après. La vie autonome doit être activée pour mettre en veille le robot. Dans ce mode, le robot ne peut pas faire d'activités mais il peut télécharger les mises à jour. La rigidité de ses moteurs est désactivée.

1. Posez votre main, 3 secondes, sur la tête de Pepper, paume sur la caméra du front et doigts sur les capteurs tactiles. Les yeux et les LEDs deviennent violets. Le robot émet un son.
2. Relâchez votre main. Pepper s'endort.

Pour le réveiller vous devez appuyer brièvement sur les capteurs tactiles de sa tête.

## 1.4 Configuration de Pepper

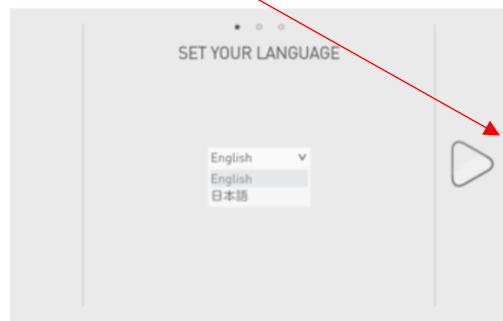
Lors du premier démarrage de Pepper, il faut configurer le robot, via la tablette.

1. Touchez l'écran



Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)

2. Choisissez la langue et cliquez sur le bouton suivant



Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)

3. Validez la licence en cochant les deux carrés et cliquez sur le bouton suivant



*Softbank robotics documentation, [doc.aldebaran.com](#)*

4. Choisissez un réseau WiFi



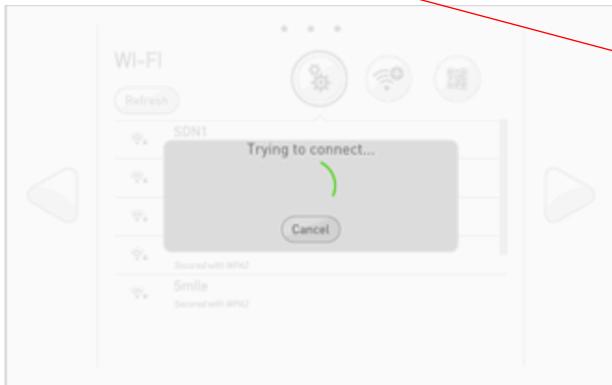
*Softbank robotics documentation, [doc.aldebaran.com](#)*

5. Complétez le mot de passe et cliquez sur le bouton Connect

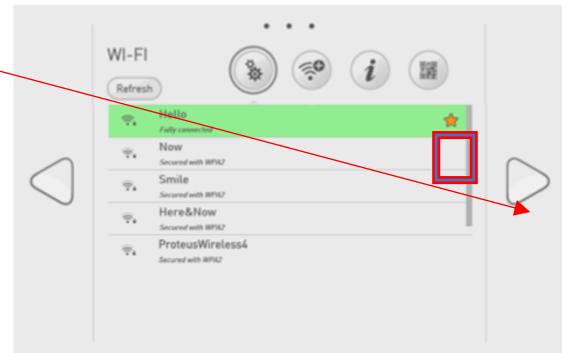


*Softbank robotics documentation, [doc.aldebaran.com](#)*

6. Attendez que la connexion se fasse. Une étoile apparaît à côté de votre connexion. Cliquez sur le bouton suivant.

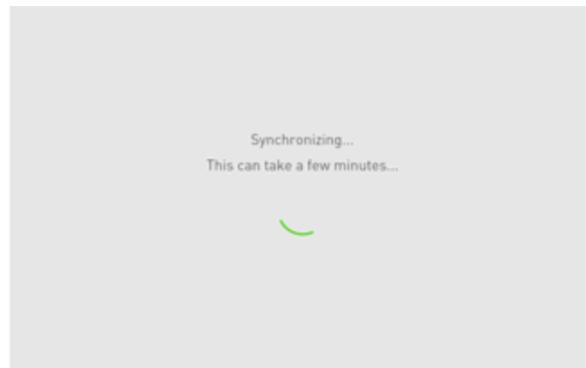


Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)



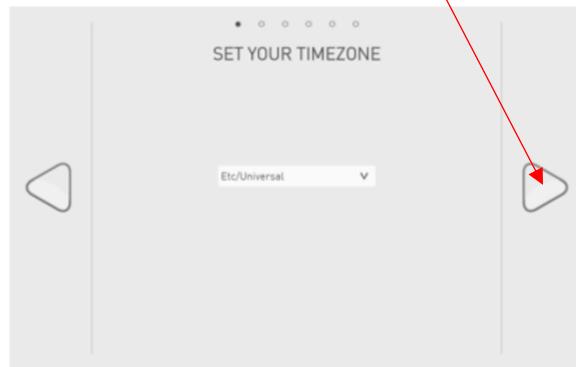
Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)

7. Patiencez quelques minutes pendant la synchronisation.



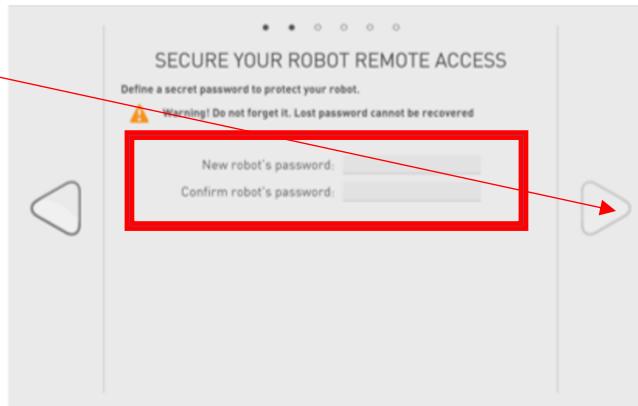
Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)

8. Définissez votre fuseau horaire et cliquez sur le bouton suivant.



Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)

9. Choisissez un mot de passe pour sécuriser l'accès à Pepper. Il vous sera demandé lorsque vous vous connecterez à sa page Web ou quand vous modifierez les paramètres sur la tablette. Cliquez sur le bouton suivant.

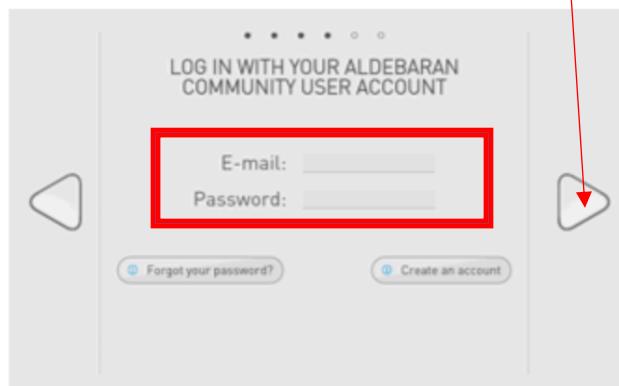


*Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)*

10. Connectez-vous à votre compte Aldebaran et cliquez sur le bouton suivant.

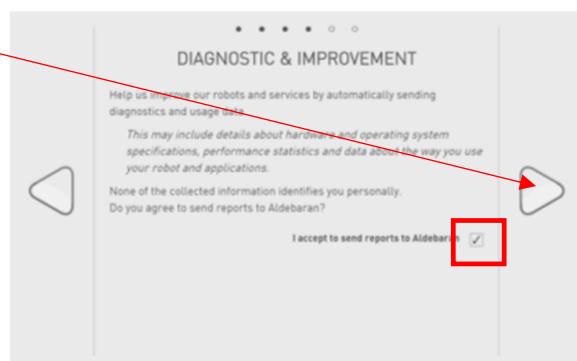
Vous pouvez télécharger des applications directement sur la tablette grâce à l'application du Store.

Attention : cette application ne fonctionne qu'avec les langues Anglais et Japonais.



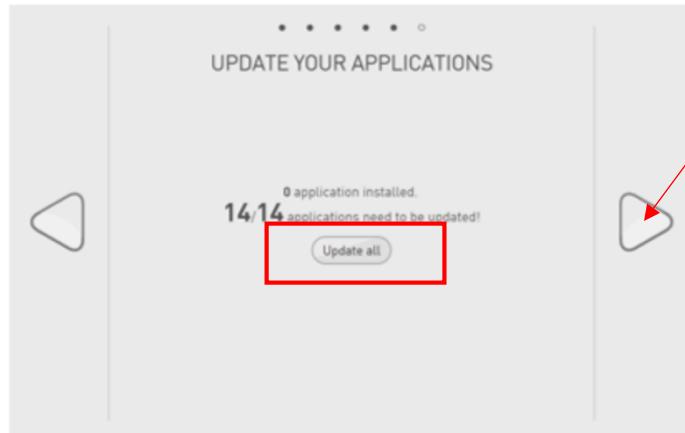
*Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)*

11. Acceptez le rapport de diagnostic automatique en cochant la petite case et cliquez sur le bouton suivant :



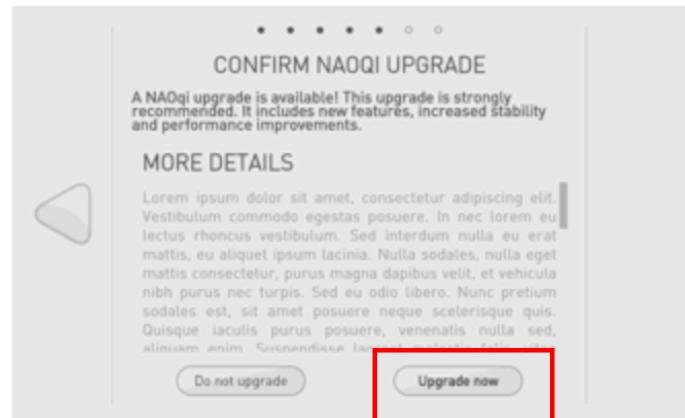
*Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)*

12. Mettez à jour les applications en appuyant sur *Update all* et cliquez sur le bouton suivant :



*Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)*

13. Mettez à jour le système NAOqi en cliquant sur *Upgrade now*.



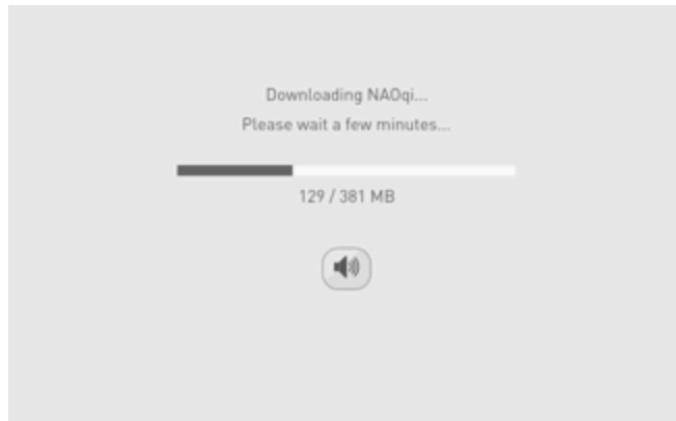
*Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)*

Un message s'affiche pour vous demander de brancher le robot si ce n'est pas déjà fait et pour vous informer de le laisser brancher durant toute la mise à jour. Cliquez sur *Start upgrade*.



*Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)*

14. Patientez quelques minutes :



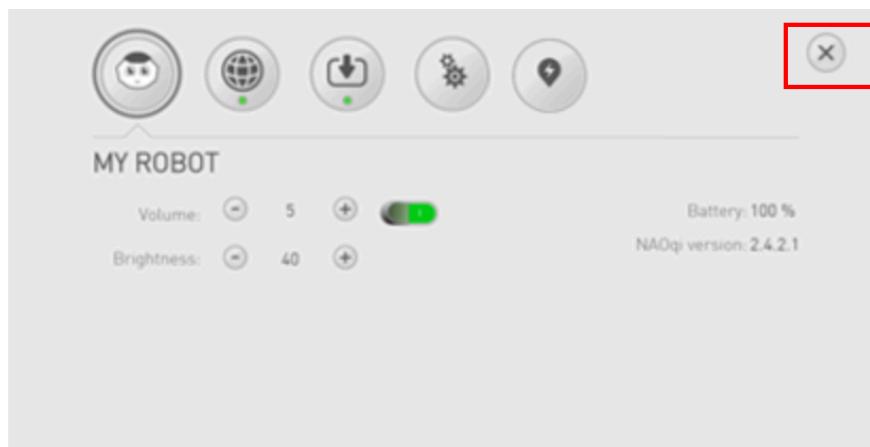
*Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)*

15. La configuration est terminée. L'écran suivant doit apparaître. Cliquez sur *Start*.



*Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)*

16. Après quelques secondes, le menu des paramètres s'affiche. Cliquez sur la croix et vous pouvez commencer à utiliser Pepper.



*Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)*

## 1.5. Menu des paramètres

- Pour accéder au menu des paramètres, appuyez sur les points de suspension, en bas à droite, de la tablette. La liste des activités disponibles apparaît. Puis appuyez sur l'icône des paramètres.



- Le menu apparaît à l'écran. La page *Mon robot* apparaît par défaut. Un cercle entoure l'icône de la page sélectionnée.
- Pour accéder à une autre page des paramètres, cliquez sur l'icône correspondant.

### 1.5.1 La page Mon robot

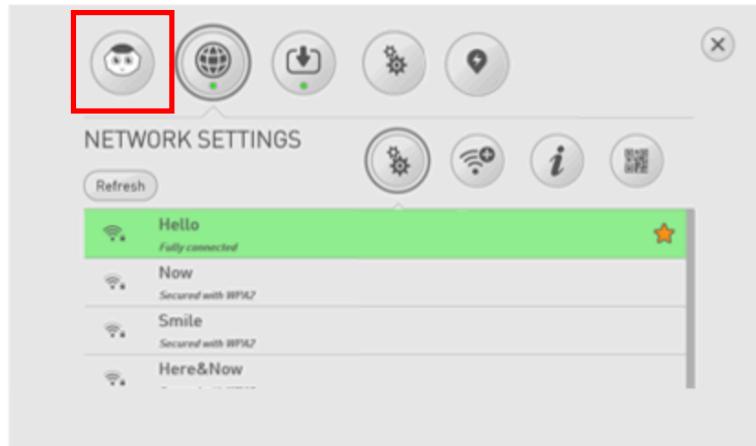
La page *Mon robot* affiche les paramètres basiques du robot et permet de les définir.



Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)

### 1.5.2 La page Paramètres réseau

La page *Paramètres réseau* permet de choisir votre connexion à un réseau.



Sofibank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)



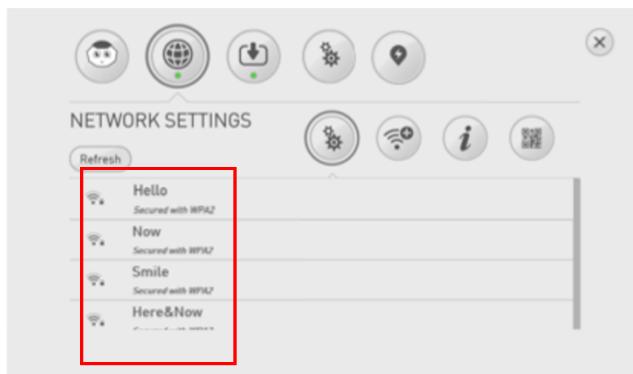
Sur cette page, on aperçoit 4 icônes :



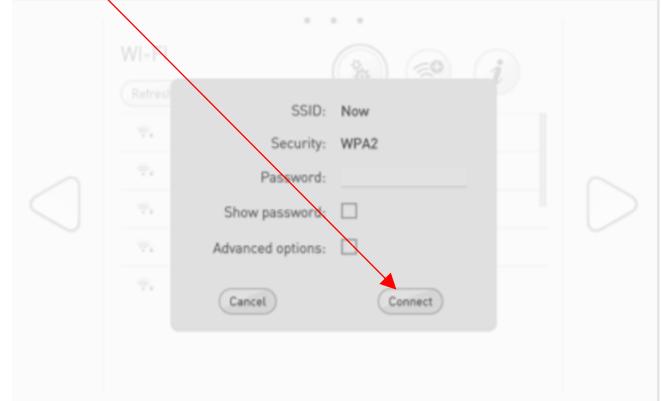
- Connexion standard au WiFi :

La page *connexion standard du WiFi* apparaît par défaut.

1. Sélectionnez un réseau WiFi
2. Complétez le mot de passe et cliquez sur le bouton **Connect**

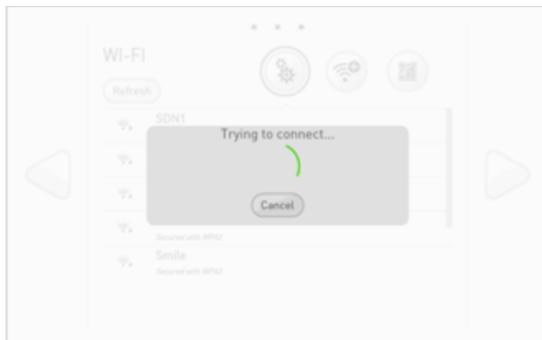


Sofibank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)

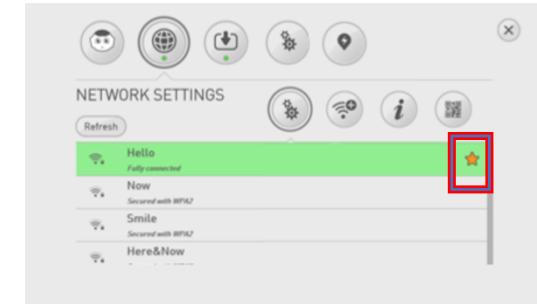


Sofibank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)

3. Attendez que la connexion se fasse. Une étoile apparaît à côté de votre connexion.

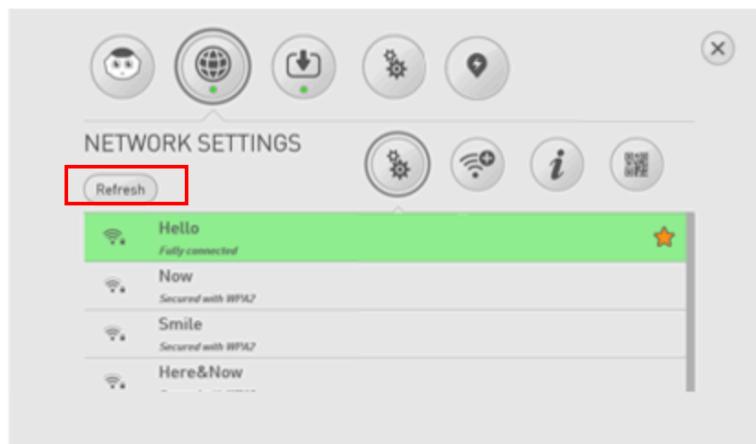


Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)



Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)

4. Si la connexion que vous souhaitez n'apparaît pas, rafraîchissez la page.



**NB :** Pour information Pepper prend en charge les connexions WPA, WPA2 et WEP. Attention, le mot de passe pour une connexion WEP40 doit être constitué de 5 caractères et pour une connexion WEP104 de 12 caractères.



- Connexion à un réseau caché ;

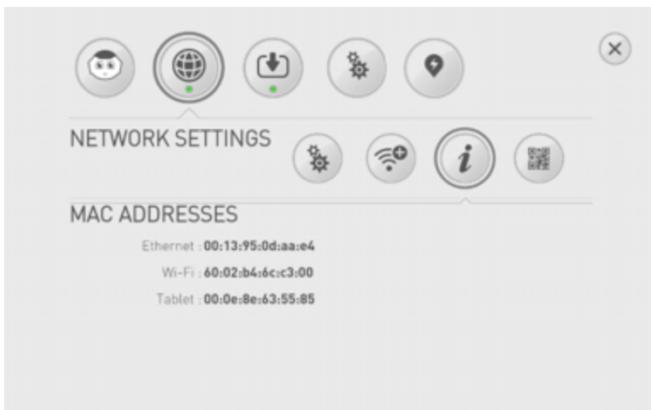


Complétez les paramètres requis et cliquez sur le bouton Connexion.

Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)



- *Section des adresses MAC ;*



*Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)*



- *Connexion WiFi avec un code QR.*



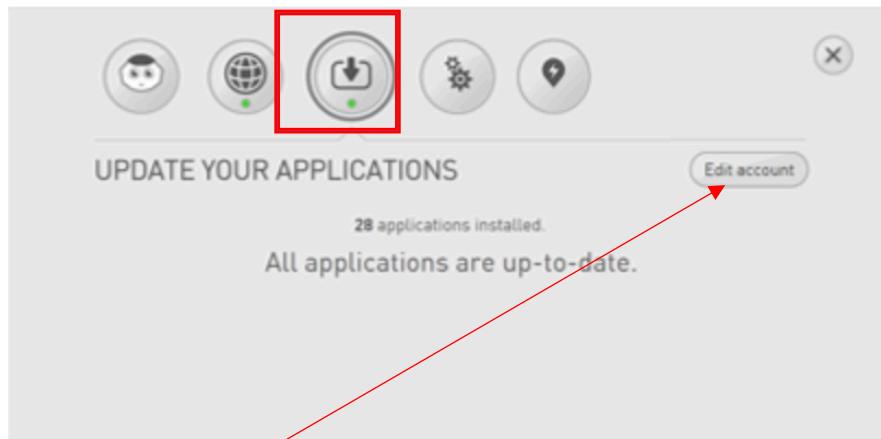
*Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)*

Si votre point d'accès WiFi nécessite de définir un filtrage, vous pourriez avoir besoin de ces informations.

### 1.5.3 La page Mise à jour des applications

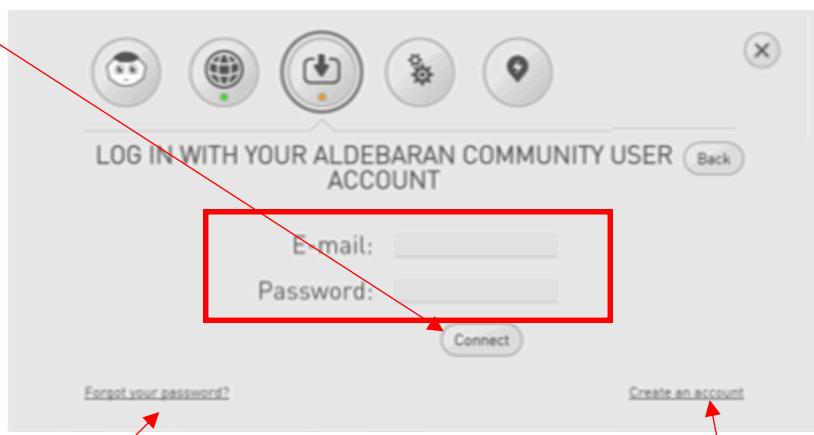
La page mise à jour des applications permet de se connecter au Cloud d'Aldebaran et de mettre à jour les applications du robot. Les mises à jour nécessitent une connexion internet.

Des applications sont à mettre jour quand une nouvelle version apparaît sur le store, quand vous téléchargez une nouvelle application sur celui-ci ou quand vous supprimez une application du robot via le logiciel Choregraphe.



*Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)*

1. Si vous n'êtes pas connecté à votre compte Aldebaran, faites-le en cliquant sur le bouton d'édition.
2. La page d'identification apparaît. Complétez les informations requises et cliquez sur le bouton Connexion. En vous connectant, vous sauvegarderez les données.

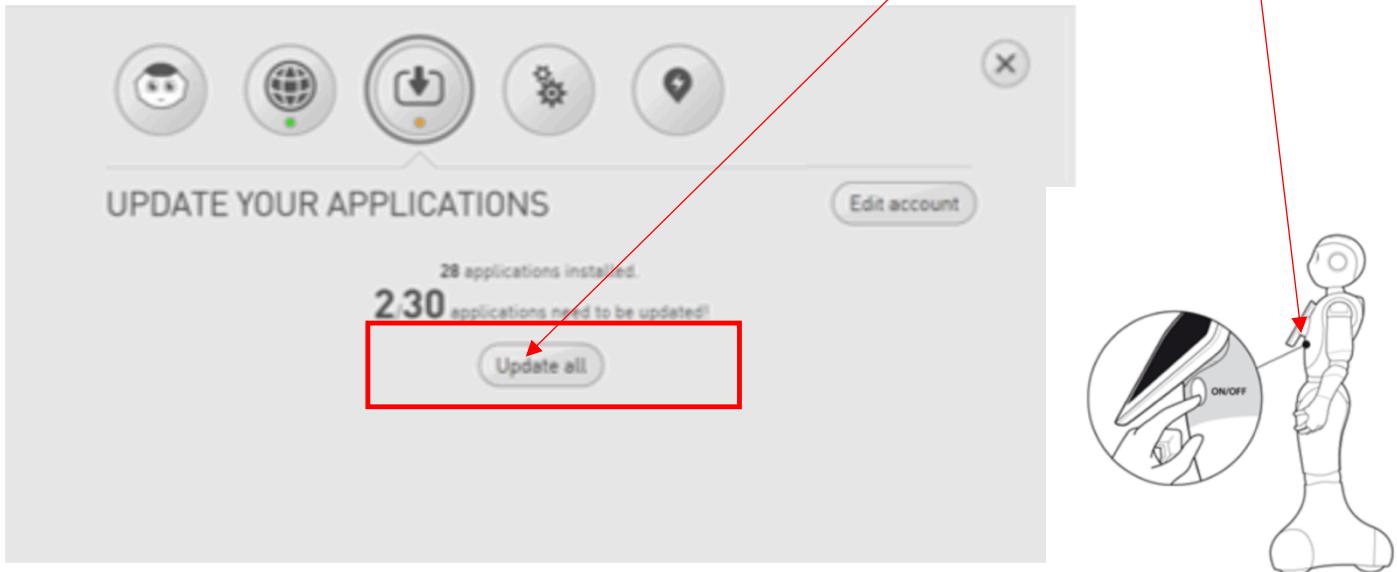


*Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)*

NB :

- Vous pouvez demander un nouveau mot de passe si vous avez oublié le tien, en cliquant sur Forgot your password.
- Vous pouvez créer un compte si vous n'en possédez pas. Cliquez sur Create an account.

3. Vous pouvez voir le nombre d'applications qui peuvent être mis à jour. Si vous souhaitez, effectuer la mise à jour, désactivez la vie autonome du robot en appuyant deux fois sur le bouton Power au centre de sa poitrine. Il soupire, arrête les activités en cours, se met en position de repos et éteint la rigidité de ses moteurs. Ensuite, cliquez sur le bouton Tout mettre à jour.



*Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)*

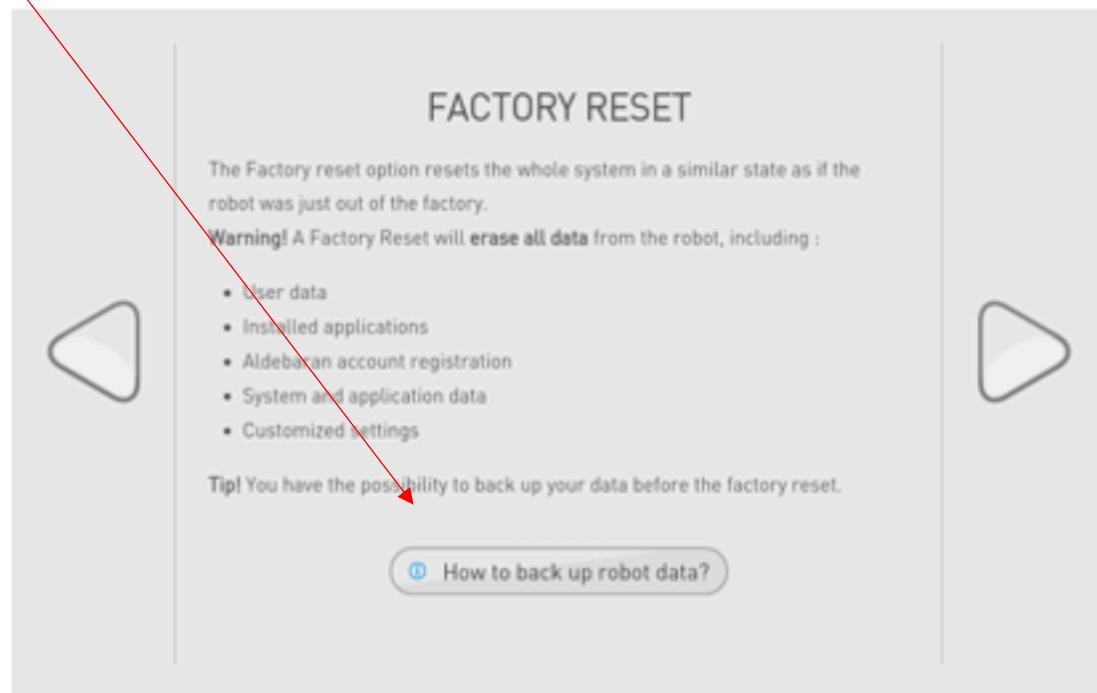
#### 1.5.4 La page Paramètres avancés

Sur cette page, vous pouvez définir les paramètres avancés du robot, tels que la langue, le fuseau horaire, le vie autonome ... Les boutons Arrêt et Redémarrer sont également affichés.

Langues du robot	Robot language: English	Robot password: ***** /	Éditer le mot de passe de Pepper
Le fuseau horaire est nécessaire pour la connexion WiFi	Timezone: Etc/Universal	Alive by default: 0	Activer ou désactiver la vie autonome au démarrage
Factory reset			Réinitialiser les paramètres d'usine. Attention, pense à sauvegarder les données de la mémoire de ton robot, via Choregraphe. Pour information, les applications téléchargées sur le store sont automatiquement gérées par le site Web. Vous ne pouvez pas les sauvegarder.

*Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)*

Lorsque vous choisissez de réinitialiser les paramètres d'usine, soyez conscient que l'ensemble du système du robot sera remis à zéro. Vous devrez par la suite, enregistrer à nouveau, installer les applications du Store, redéfinir les paramètres et restaurer la mémoire du robot si tu le souhaites. Cliquez sur le bouton [\*How to back up robot data\*](#) pour avoir des informations sur la sauvegarde et la restauration des données.



*Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)*

## 1.6. Vie autonome

Par défaut, Pepper est réactif. Il oriente sa tête vers les humains, il réagit aux sons, aux mouvements et aux contacts tactiles. Il est même capable de dialoguer avec vous.

### 1.6.1 Parler avec le robot :

1. Approchez-vous du robot aux alentours d'un mètre. Assurez-vous qu'il vous regarde. Ses yeux doivent devenir bleus. Dans ce cas, il est à votre écoute.
2. Dites une phrase courte, pré-enregistrée dans le robot, en une seule fois. Pour les connaître, référez-vous à la liste *Basic channel*, ci-dessous. S'il vous a entendu ses yeux deviennent verts. Il traite l'information.

3. Le robot vous répond. Ses yeux sont blancs. Quand Pepper a les yeux de cette couleur, il n'est plus capable de détecter les mouvements ou d'écouter.
4. Ses yeux redeviennent bleus. Il est à nouveau à votre écoute. Vous pouvez lui parler.

Il est possible qu'il n'ait pas compris. Attendez que les yeux redeviennent bleus et répétez la phrase. Si le robot, vous coupe la parole, c'est qu'il a interprété une pause comme la fin de la phrase. Pour éviter ces malentendus, parlez fort de manière continu, en insistant sur l'articulation et en regardant le robot. Privilégiez un endroit calme.

Grâce à cette fonctionnalité de dialogue, vous pouvez demander à Pepper de lancer une activité :

1. Dites « Démarrer » suivi du nom de l'application.
2. Une phrase de déclenchement est parfois nécessaire.

Par exemple, pour lancer l'application *Jeu étoiles*, vous dîtes « Démarrer jeu étoiles » alors que pour lancer l'application *Danse*, vous devez dire « Danse Pepper ».

### 1.6.2 Basic channel : Que puis-je dire à Pepper ?

Les phrases suivantes vont vous permettre d'interagir avec Pepper. Ce sont les phrases types. Il est possible que certaines variations de mots puissent fonctionner. Le robot ne peut comprendre qu'une seule langue à la fois. Vous devez lui demander si vous voulez communiquer dans une autre langue, ou alors, vous pouvez le paramétrier dans une autre langue via la tablette, ou via *Choregraphe*.

## Faire les présentations

<i>Français</i>	<i>Anglais</i>
Comment ça va ?	How are you ?
Fais-moi coucou.	Can you say goodbye ?
Que sais-tu faire ?	What can you do ?

Quelles sont tes applications ?	Tell me all you can do.
Quelles langues parles-tu ?	What languages do you speak ?
Comment tu t'appelles ?	What's your name ?
Quel est ton surnom ?	What is your nickname ?
Présente-toi.	Introduce yourself.
Quelle taille fais-tu ?	How tall are you ?
Combien pèses-tu ?	How much do you weigh ?
Quel âge as-tu ?	How old are you ?
Quand est ton anniversaire ?	When is your birthday ?
Es-tu un garçon ou une fille ?	Are you a boy or a girl ?
Tu es en quelle matière ?	What are you made of ?
De quelle couleur es-tu ?	What colour are you ?
Pourquoi tu as une tablette ?	What is your tablet for ?
Es-tu gentil ?	Are you kind ?
Es-tu cool ?	Are you cool ?
Es-tu intelligent ?	Are you intelligent ?
Est-ce que je peux te faire confiance ?	Can I trust you ?
Es-tu humain ?	Are you human ?
Es-tu vivant ?	Are you alive ?
Es-tu conscient ?	Are you sentient ?
Est-ce que tu peux penser ?	Can you think ?
Peux-tu ressentir des émotions ?	Can you feel emotions ?
Comment détectes-tu les émotions ?	How do you detect emotions ?
Est-ce que tu manges ?	Do you eat ?
Est-ce que tu peux parler ?	Can you talk ?
Est-ce que tu peux me voir ?	Can you see me ?
Est-ce que tu peux m'entendre ?	Can you hear me ?
Est-ce que tu peux me comprendre ?	Can you understand me ?
Est-ce que tu peux me reconnaître ?	Can you remember me ?
Es-tu un robot ?	Are you a robot ?

C'est quoi un robot ?	What is a robot ?
C'est quoi un robot humanoïde ?	What is a humanoid robot ?
Est-ce que les robots vont remplacer les humains ?	Will robot replace humans ?
Connais-tu Nao ?	Do you know Nao ?
Connais-tu Roméo ?	Do you know Romeo ?
Connais-tu les lois de la Robotique ?	Do you know the laws of robotics ?
Pourquoi tu t'appelles Pepper ?	Why is your name Pepper ?
Qui est-ce qui t'a appelé comme ça ?	Who decided to call you like that ?
Qui t'a fabriqué ?	Who made you ?
Où as-tu été fabriqué ?	Where were you made ?
As-tu de la famille ?	Do you have a family ?
As-tu des amis ?	Do you have friends ?
Comment s'appelle ton ami ?	What is your friend's name ?
Je peux être ton ami ?	Can we be friends ?
Est-ce que tu sais parler anglais ?	-
-	Can you speak French ?
Est-ce que tu sais parler japonais ?	Can you speak Japanese ?
Est-ce que tu sais parler chinois ?	Can you speak Chinese ?

## Demander des informations

Quelle est ton niveau de batterie ?	What is your battery level ?
Es-tu en charge ?	Are you charging ?
Quelle est ton adresse IP ?	What is your IP address ?
Es-tu connecté à internet ?	Are you connected to internet ?
Quelle heure est-il ?	What time is it ?
Quel jour on est ?	What day are we ?
Qu'est-ce que j'ai dit ?	What did I say ?
Tu peux répéter ?	Can you repeat please ?

## Donner un ordre

Parle anglais.	-
-	Speak French.
Parle japonais.	Speak Japanese.
Parle chinois.	Speak Chinese.
Parle moins fort.	Speak softer.
Parle plus fort.	Speak louder.
Bouger la tête.	Can you move your head ?
Tourne la tête à droite.	Can you turn your head to the right?
Tourne la tête à gauche.	Can you turn your head to the left.
Lève la tête.	Can you move your head ?
Regarde-moi.	Can you look at me ?
Regarde ailleurs.	Can you look elsewhere ?
Baisse les yeux.	Stop looking at me.
Tends les bras.	Can you stretch your arms ?
Tends le bras gauche.	Can you stretch your left arm ?
Tends le bras droit.	Can you stretch your right arm ?
Lève les bras.	Raise your arms.
Fais bouger tes mains.	Make your hands move.
Fais bouger la main gauche.	Make your left hand move.
Fais bouger la main droite.	Make your right hand move.
Peux-tu ouvrir les mains ?	Can you open your hands ?
Peux-tu ouvrir la main gauche ?	Can you open your left hand ?
Peux-tu ouvrir la main droite ?	Can you open your right hand ?
Peux-tu fermer les mains ?	Can you close your hands ?
Peux-tu fermer la main gauche ?	Can you close your left hand ?
Peux-tu fermer la main droite ?	Can you close your right hand ?
Mets tes mains en arrière.	Can you put your hand behind ?
Mains en l'air !	Hands up !

### 1.6.3 Autres fonctionnalités de Basic Channel :

D'autres réactions du mode autonome :

- Quand vous couvrez la caméra du front il n'est pas content.
- Il aime bien quand vous lui caressez la tête.
- S'il tombe, il se relève tout seul.
- Il vous demandera de l'aide, s'il se trouve dans une position inconfortable...

### 1.6.4 Désactiver la vie autonome :

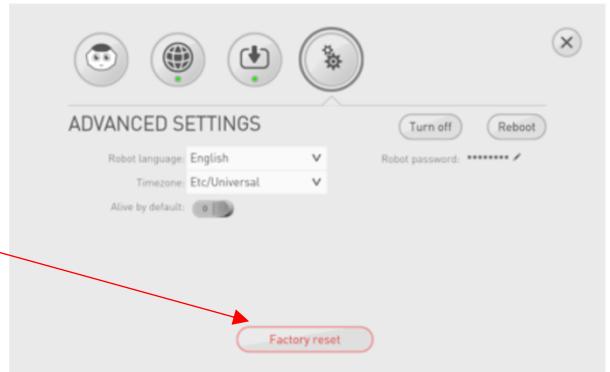
Vous pouvez désactiver la vie autonome pour tester un nouveau programme créé sur Choregraphe. Appuyez deux fois sur le bouton on/off situé sur la poitrine du robot. Le robot soupire et arrête ses activités. Il se met en posture de repos et désactive la rigidité de ses moteurs.

Pour le réactiver vous devez appuyer à nouveau deux fois sur le bouton on/off. Le robot émet un son joyeux et reprend sa vie.

Vous pouvez également activer/désactiver la vie autonome sur Choregraphe en appuyant sur le cœur de la barre d'outils. Lorsqu'il est bleu, la vie autonome est activée.



Vous pouvez modifier le statut par défaut, de l'activation de la vie autonome. Pour cela allez dans les paramètres avancés et cliquez sur le bouton [Alive by default](#).



Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)

### 1.6.5 Améliorer la vie autonome :

Grâce au logiciel *Choregraphe*, vous allez pouvoir créer des activités solitaires et interactives, les installer dans la mémoire du robot et ainsi améliorer sa vie autonome. Référez vous au paragraphe *Création d'un comportement autonome*.

## 1.7. Fermeture d'une activité

1. Posez votre main sur la tête de Pepper pendant 1 seconde, paume sur la caméra du front et doigts sur les capteurs tactiles. Les LEDs des yeux et des épaules deviennent verts. Le robot émet le son « BLOP ».
2. Tirez votre main. L'activité s'arrête et la tablette affiche la page d'accueil.

## 1.8. Les voyants d'état

Il est possible que Pepper émette des notifications. Dans ce cas, les LEDs des épaules clignotent. Un code couleur existe.

Voyant vert : feed-back sur une action en cours. Aucune gravité.

Voyant orange : avertissement d'un problème rencontré. Des fonctionnalités peuvent dysfonctionnées. Il faut résoudre le problème pour que le robot fonctionne correctement. Gravité moyenne.

Voyant rouge : erreur du système. Le robot devient inutilisable. Il faut agir rapidement.

Exemple de notification : « Je ne peux plus bouger. J'ai détecté une erreur sur un de mes appareils vitaux. Veuillez essayer de me redémarrer pour résoudre le problème. (LEDs rouge)

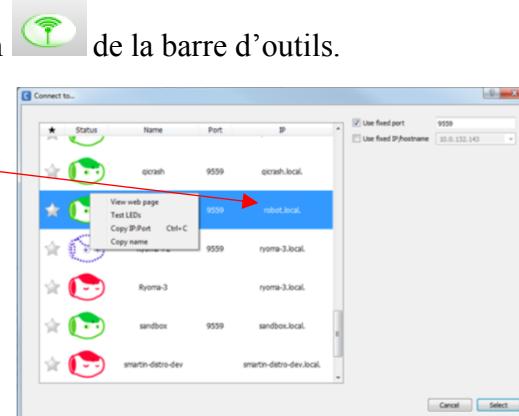
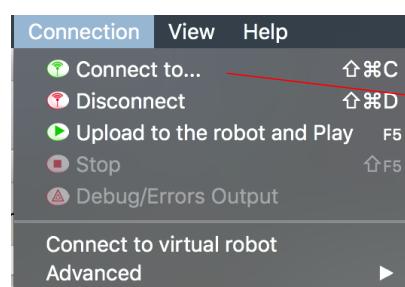
## 1.9. Page web de Pepper

### 1.9.1 Accéder à la page Web du robot sur un navigateur web :

1. Allumez le robot.
2. Connectez le robot à votre ordinateur grâce au WiFi. Pour cela, allez dans les paramètres réseaux de votre robot et dans ceux de votre ordinateur. Vous pouvez utiliser un routeur WiFi.
3. Ensuite, appuez brièvement sur le bouton on/off du robot. Il va énoncer son adresse IP.
4. Tapez la dans la barre d'adresse d'un navigateur Web. Une fenêtre d'authentification apparaît.
5. Complétez l'identifiant et le mot de passe.

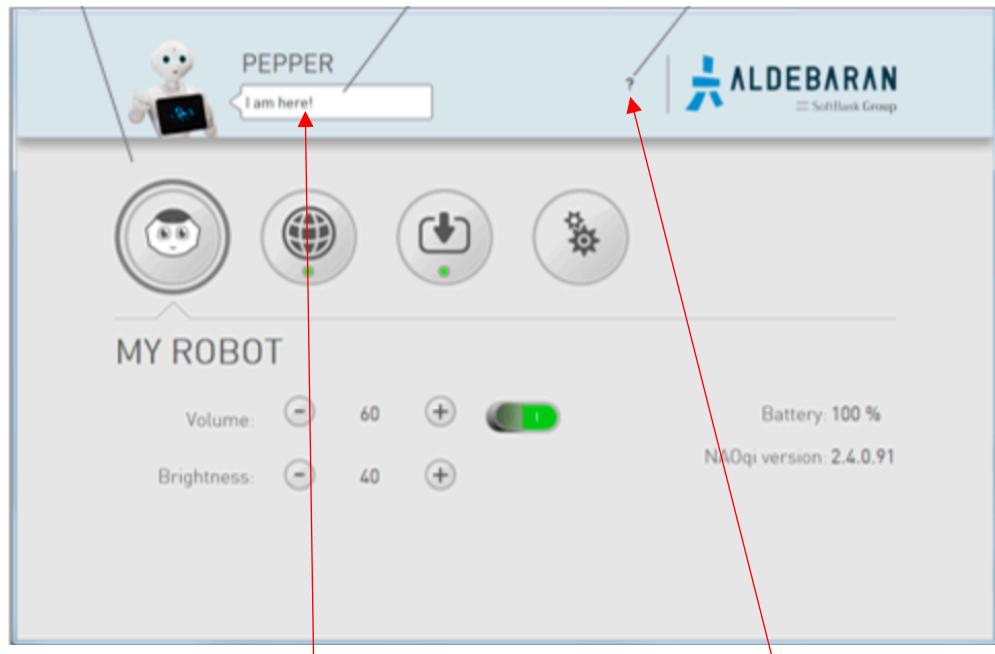
### 1.9.2 Accéder à la page Web du robot sur Choregraphe :

1. Allumez le robot.
  2. Connectez le robot à votre ordinateur grâce au WiFi. Pour cela, allez dans les paramètres réseaux de votre robot et dans ceux de votre ordinateur. Vous pouvez utiliser un routeur WiFi.
  3. Démarrez le logiciel Choregraphe.
  4. Cliquez sur *Connection*, puis sur *Connect to* ou appuyez sur le bouton  de la barre d'outils.
- La page connexion apparaît.
5. Sélectionnez le robot de votre choix et cliquez droit dessus.
  6. Choisissez « View web page ».



Sofbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)

**NB :** Si vous avez perdu le mot de passe de l'interface Web, vous devez réinitialiser les paramètres d'usine, en allant dans les paramètres avancés et en cliquant sur le bouton *Reset Factory*.



Sofbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)

La page Web vous permet d'accéder aux paramètres du robot, d'ouvrir [l'assistant de démarrage ou la documentation en ligne](#) et de [faire dire des phrases courtes](#) à Pepper. Pour qu'il dise la phrase que vous tapez, vous devez appuyer sur la touche *Entrée*.

## 1.10. Téléchargement d'une application sur le Store Aldebaran

### 1.10.1 Sur la tablette du robot :

1. Malheureusement l'application *Store* ne fonctionne pas avec la langue française. Paramétrez votre robot en Anglais ou en Japonais via les paramètres avancés.
2. Assurez-vous que votre robot soit connecté à votre compte Aldebaran. Le cas échéant, connectez-vous en allant dans les paramètres avancés, sur la page *Mise à jour des applications*.
3. Appuez sur *Démarrer*.
4. Appuez sur l'application *Store*.
5. Choisissez la rubrique *Applications-Pepper*.
6. Cliquez sur l'application de votre choix.
7. Acceptez les conditions d'utilisation.
8. Cliquez sur le bouton *Install*. L'application sera installée lors de la prochaine mise à jour de Pepper.

### 1.10.2 Sur un ordinateur :

1. Grâce à un ordinateur connecté à internet, accédez au site <https://store.aldebaran.com> .
2. Cliquez sur le bouton *Sign in*. La page de connexion apparaît.
3. Notez votre identifiant et votre mot de passe, puis appuyez sur *Se connecter*. Si vous avez oublié votre mot de passe, appuyez sur *Mot de passe oublié*. Si vous n'avez pas de compte, il faut vous en créer un en cliquant sur *Créer un compte*, puis en remplissant le formulaire d'inscription.
4. Choisissez la rubrique *Applications-Pepper*.
5. Cliquez sur l'application de votre choix.
6. Acceptez les conditions d'utilisation.
7. Cliquez sur le bouton *Install*. L'application sera installée lors de la prochaine mise à jour de Pepper.

## 2 : Utilisation du logiciel *Choregraphe*

### 2.1 Présentation du logiciel

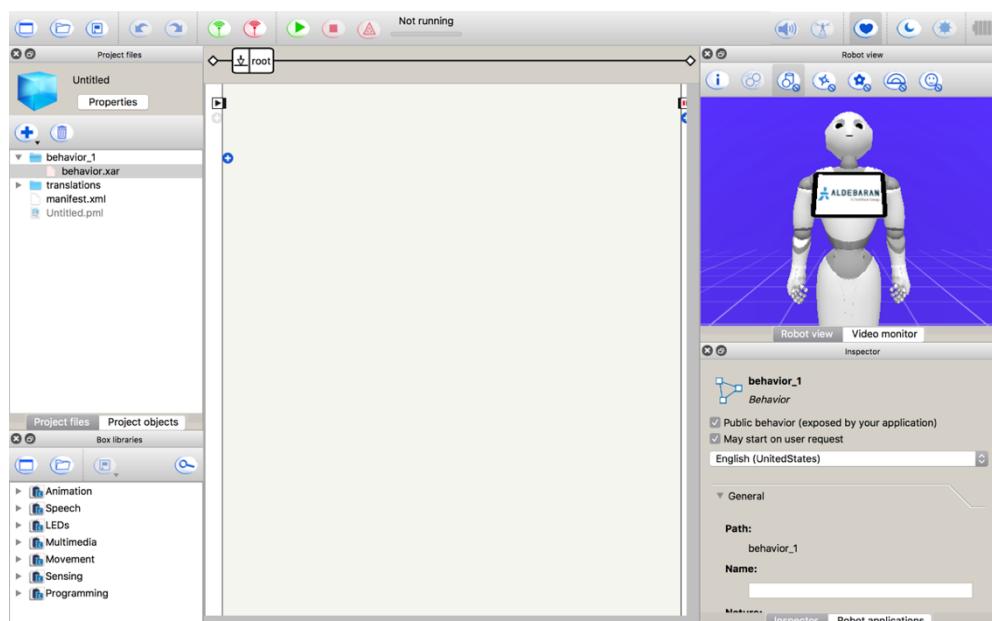
*Choregraphe* est un logiciel à installer sur l'ordinateur. Il permet de créer des applications complexes sans écrire une seule ligne de code et de les tester sur un robot virtuel ou sur un robot réel. Il offre la possibilité de contrôler PEPPER. Il permet également de modifier des boîtes déjà intégrés au logiciel en y intégrant son propre code Python.

### 2.2 Installation

1. Téléchargez la dernière version de la suite *Choregraphe*, en fonction de votre système d'exploitation, sur le site <https://community.aldebaran.com/en/resources/software>. Vous devez vous identifier pour accéder à la page.
2. Démarrez l'installation, acceptez le contrat de licence et suivez les étapes.
3. Ouvrez *Choregraphe* et activez la licence en entrant la clé valide. Si vous n'en avez pas, vous pouvez utiliser la version d'essai pendant 80 jours.

### 2.3 Interface

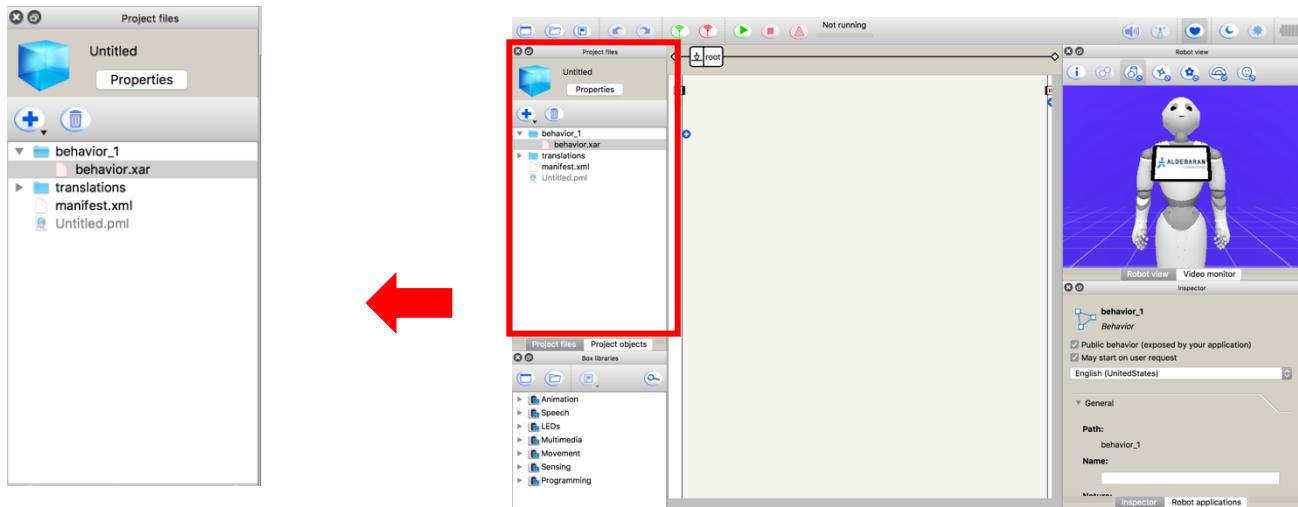
Voici l'interface, par défaut, du logiciel *Choregraphe* v2.5.5.



### 2.3.1. Les fenêtres affichées par défaut

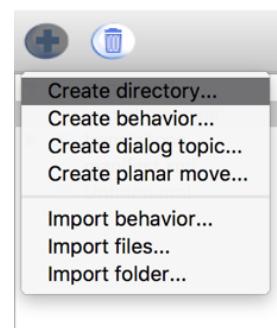
Dans la colonne de gauche, vous pouvez trouver :

A. La fenêtre **Project files** : ce sont les éléments qui se retrouvent dans ton projet. Les musiques, les images et les vidéos que vous insérez dans l'application s'afficheront ici. Cette fenêtre vous permet également d'édition des propriétés du projet.



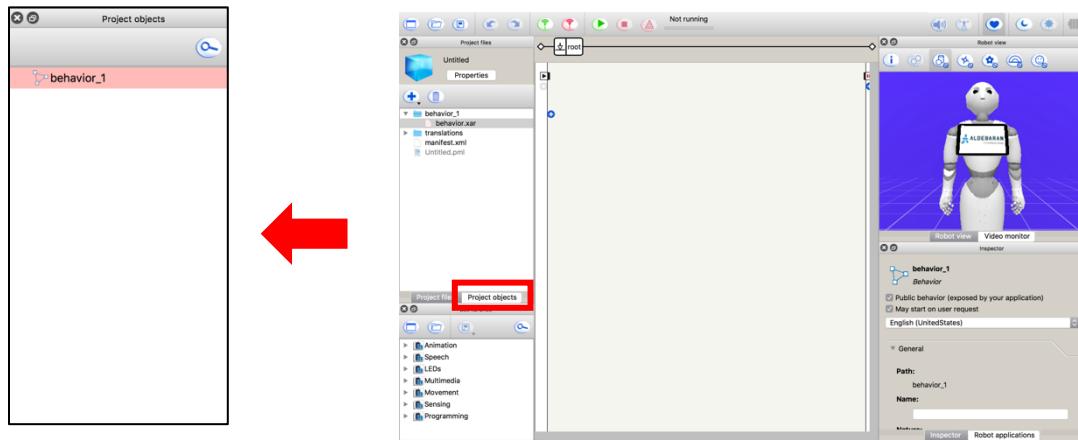
- Pour créer un nouveau dossier (*Create directory*) ou importer un fichier (*Import files*)

cliquez sur . Le menu déroulant suivant apparaît :

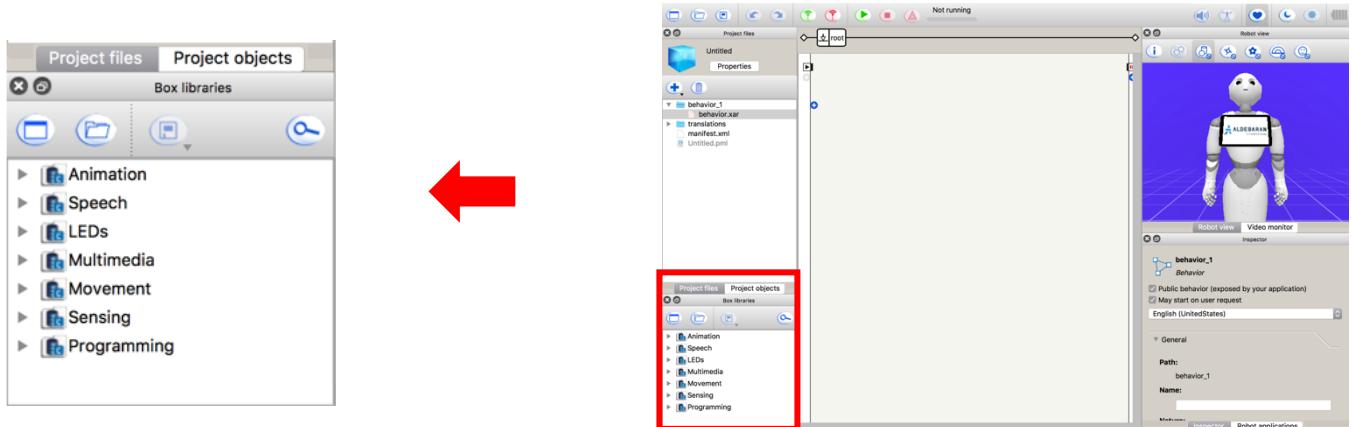


- Pour supprimer un dossier ou un fichier, sélectionnez-le et appuyez sur .
- Pour accéder aux propriétés du projet cliquez sur .

B. La fenêtre **Project objects** : vous pouvez parcourir et modifier un comportement de votre projet grâce à cette fenêtre. Vous pouvez utiliser la fonction recherche . Cette fenêtre est réduite par défaut. Pour y accéder, vous devez cliquer sur l'onglet *Project objects*.

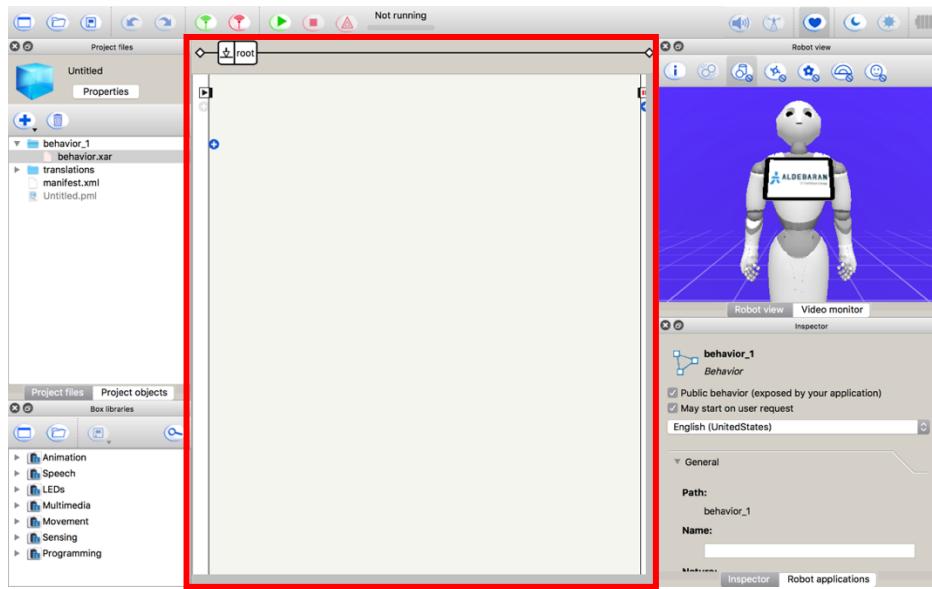


C. La fenêtre **Box libraries** : vous y trouvez toutes les actions (boîtes) que vous pouvez ajouter dans votre espace de programmation. Elle est constituée de plusieurs listes qui classent les boîtes de comportement dans des dossiers.

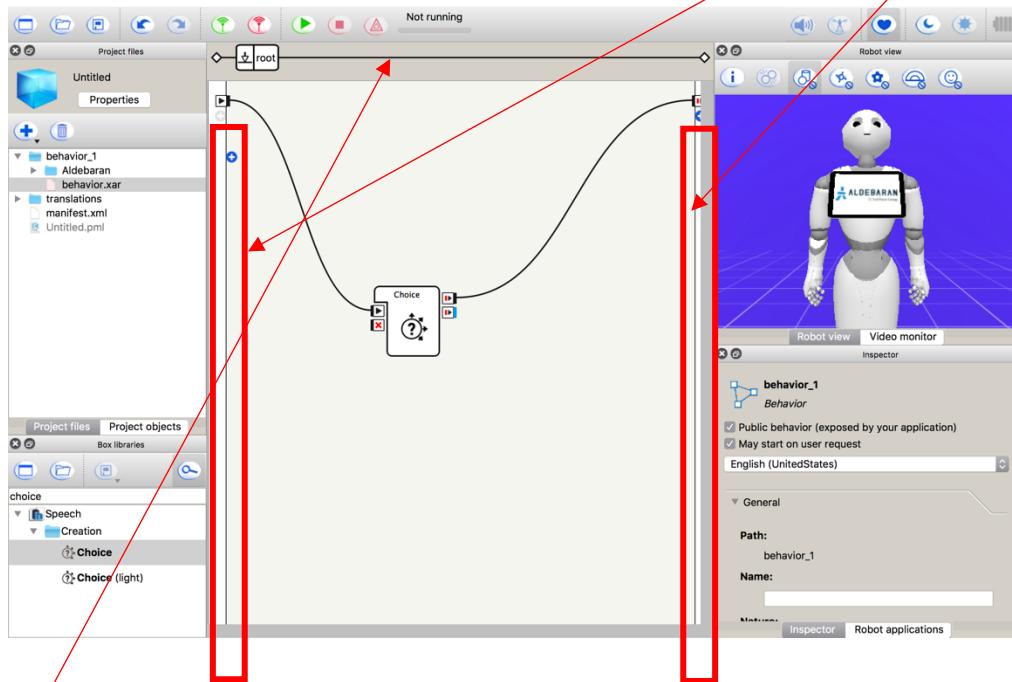


- Pour créer une nouvelle liste appuyez sur  et pour enregistrer les modifications apportées à celle-ci appuyez sur .
- Utilisez la fonction recherche pour accéder directement à une boîte dont vous connaissez le nom. Pour cela, cliquez sur .

D. La fenêtre au centre est ton **espace de travail**, l'endroit où tu vas faire la programmation.



Vous pouvez sélectionner des boîtes dans la colonne de gauche et de droite, puis les déposer dans l'espace de travail. Ensuite, vous pouvez lier les boîtes aux bordures d'entrée et de sortie.



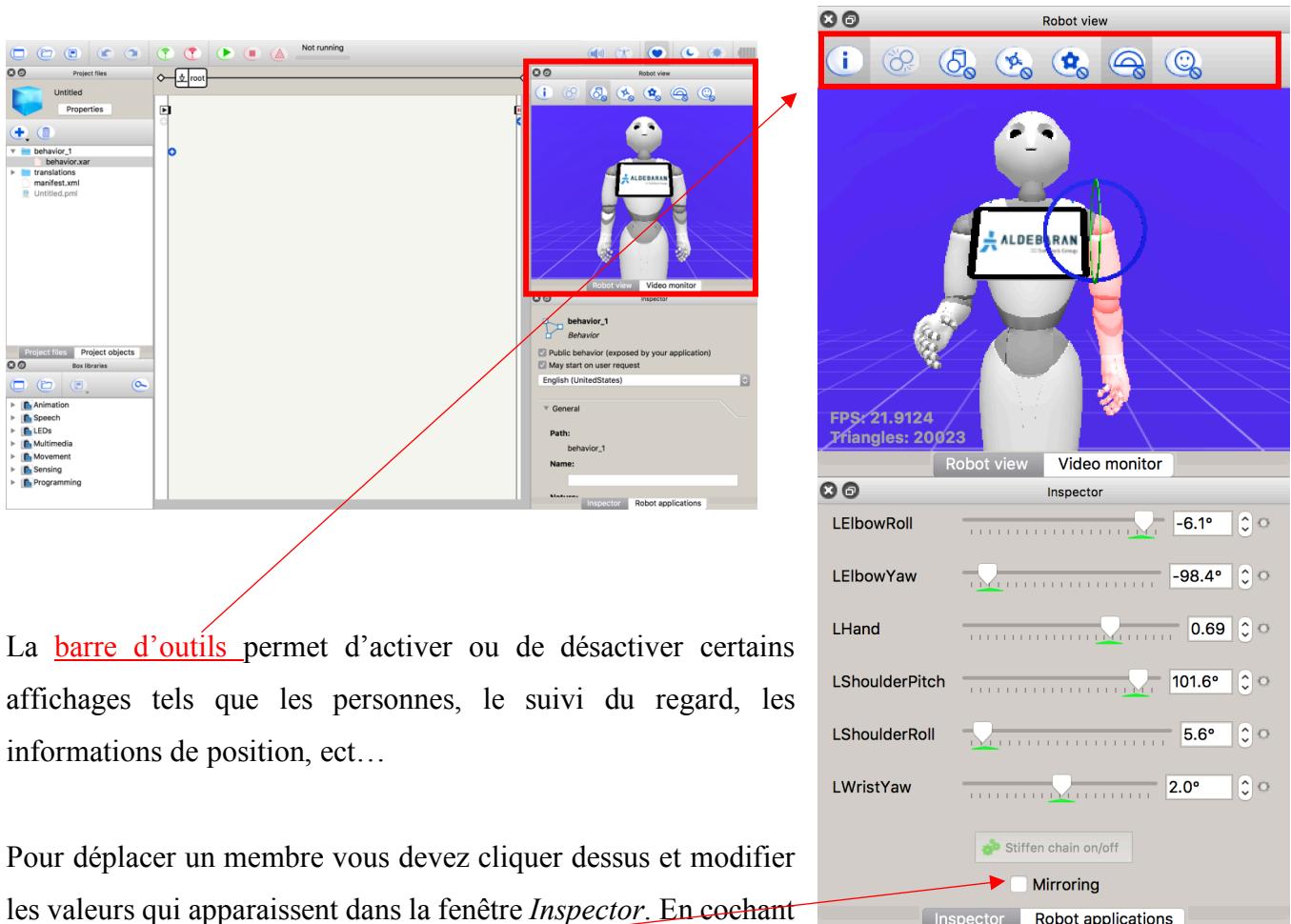
Le chemin de programmation permet de parcourir les différents niveaux de ton projet. Cliquez sur



pour revenir au niveau principal.

Dans la colonne de droite, vous pouvez trouver :

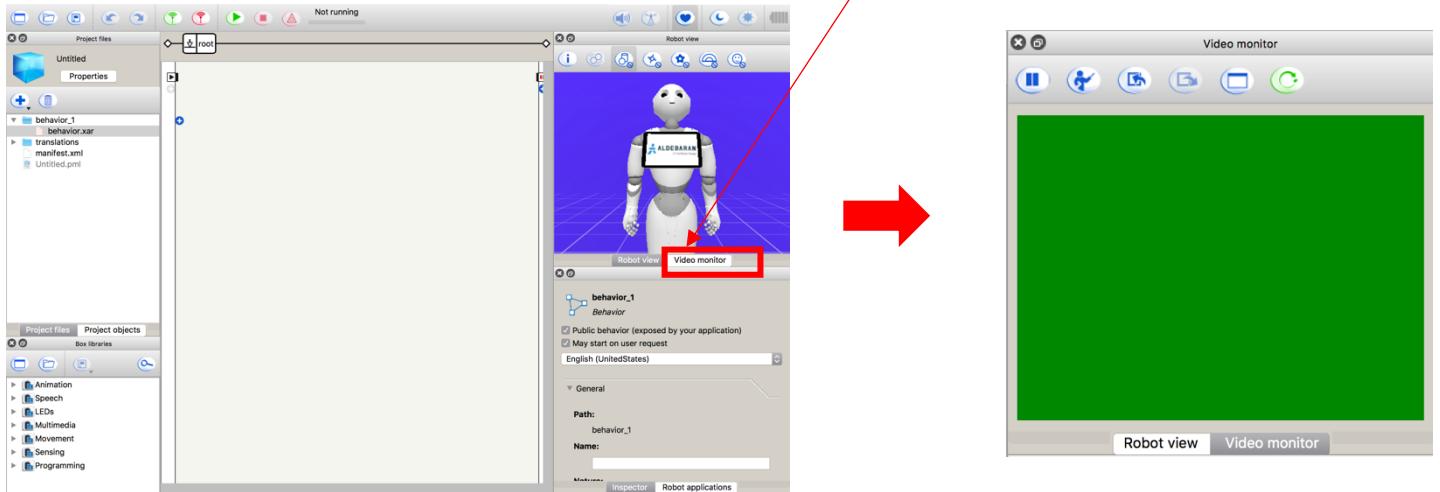
E. La fenêtre **Robot view** : Vous pouvez voir le robot, qui est connecté, faire les mouvements que vous avez programmés dans votre espace de travail (simulation). Cette fenêtre vous permet aussi de déplacer les membres du robot, de vérifier les mesures de position et d'afficher les phrases dites ou entendues par le robot, ainsi que les objets, personnes, ect...



Pour déplacer un membre vous devez cliquer dessus et modifier les valeurs qui apparaissent dans la fenêtre *Inspector*.

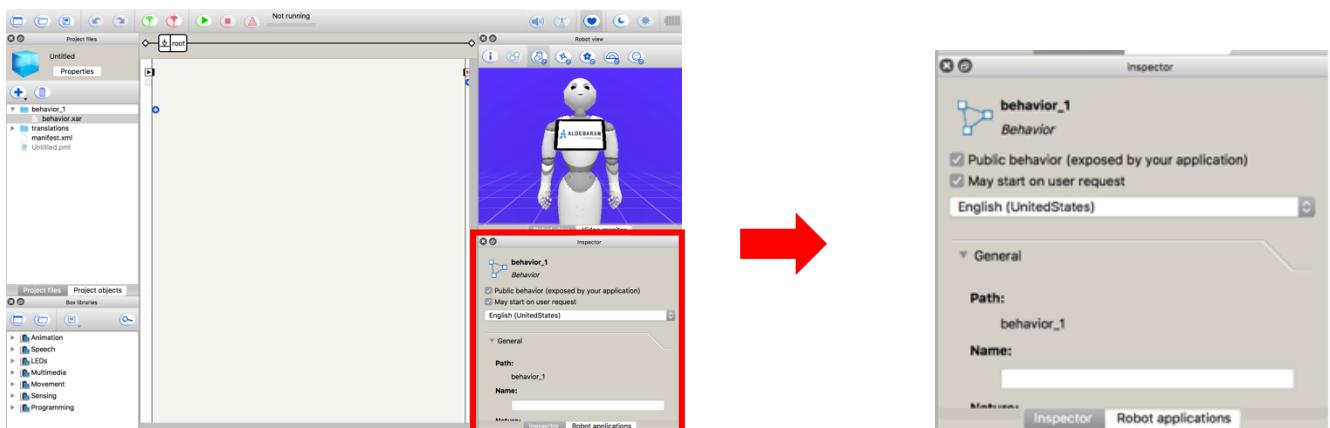
En cochant Mirroring, vous ferez bouger les deux membres de la même manière.

F. La fenêtre **Video monitor** : Elle vous donne accès à la caméra du robot et montre ce que voit le robot en temps réel. Pour y accéder, cliquez sur l'onglet **Video monitor**.

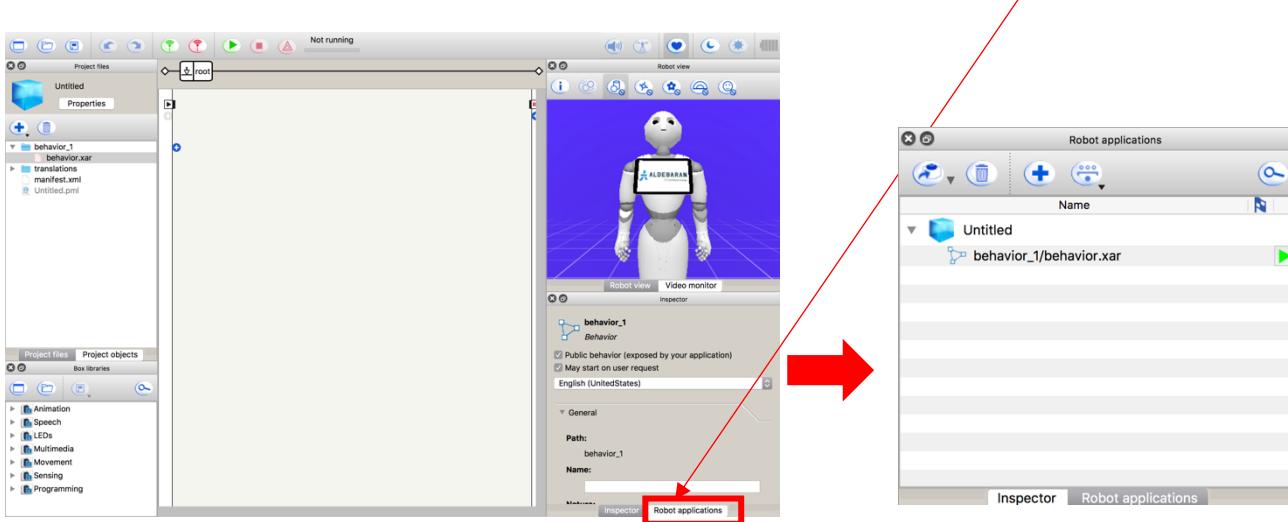


- Pour interrompre la lecture de la vidéo appuez sur 
- Pour capturer une image afin d'enregistrer un visage ou un objet, dans la base de données de reconnaissance de vision, appuez sur .
- Pour importer une base de données de reconnaissance de vision, appuez sur  et pour exporter cliquez sur .
- Pour enregistrer la base de données dans Pepper connecté appuez sur . Attention, cela remplace l'ancienne.

G. La fenêtre **Inspector** : Elle affiche les propriétés des éléments sélectionnés tels que les caractéristiques d'une boîte ou les données sur la position du robot.



H. La fenêtre ***Robot applications*** : elle recense toutes les applications installées dans la mémoire du robot connecté. Pour y accéder, cliquez sur l'onglet ***Robot applications***.

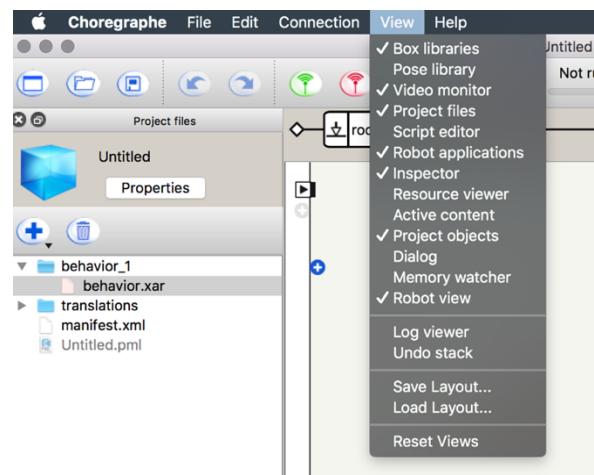


- Démarrez ou arrêtez les applications en appuyant sur le bouton
- Si vous souhaitez supprimer une application, sélectionnez-la et appuyez sur
- Installez l'application dans le robot en cliquant sur
- Ce bouton permet de régler les affichages dans cette fenêtre et celui-ci permet de faire une recherche.

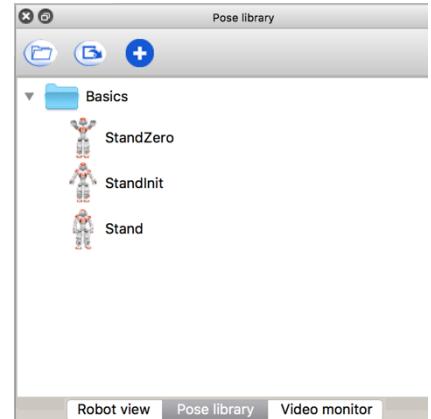
### 2.3.2. Personnalisation de l'interface

Il existe d'autres fenêtres qui ne sont pas affichées par défaut. Vous pouvez les faire apparaître et personnaliser votre interface.

- Cliquez sur *View* dans la barre de menu ;
- Choisissez les fenêtres que vous souhaitez faire apparaître ;
- Glissez-les aux endroits désirés.



F. La fenêtre **Pose library** : Elle est affichée par défaut. Vous y trouvez plusieurs poses que le robot peut faire. Cliquez directement dessus pour effectuer la position ou alors glissez-les dans la zone de programmation pour vous en servir comme les boîtes de la bibliothèque.



G. La fenêtre **Script editor** : vous pouvez afficher et éditer des scripts de boîte. La syntaxe de la programmation est colorée afin d'identifier facilement les différentes fonctions.

```

1  # /!\ Generated content. Do not edit!
2  class MyClass(GeneratedClass):
3      def __init__(self):
4          try: # disable autoBind
5              GeneratedClass.__init__(self, False)
6          except TypeError: # if NAOqi < 1.14
7              GeneratedClass.__init__(self )
8
9          self.tts = ALProxy("ALTextToSpeech")
10         self.sentences = {
11             "Arabic" : " هل تقب اليوكولاده؟",
12             "Czech" : " Máš rád čokoládu? ",
13             "Danish" : " Kan du lide chokolade? ",
14             "German" : " Mögen Sie Schokolade? ",
15             "Greek" : " ",
16             "English" : " Do you like chocolate? ",
17             "Spanish" : " ¿Te gusta el chocolate?",
18             "Finnish" : " Pidätkö suklaasta? ",
19             "French" : " Aimes-tu le chocolat ? ",
20             "Italian" : " Ti piace il cioccolato? ",
21             "Japanese" : " チョコレートが好きですか？",
22             "Korean" : " 당신은 초콜릿을 좋아합니까? ",
}

```

H. La fenêtre **Dialog** : Elle permet de visualiser l'ensemble des interactions verbales. Vous pouvez également parler au robot connecté grâce à la barre de texte.



Vous avez également des fenêtres qui vous permettent de visualiser les données du robot, de gérer la mémoire, d'afficher les notifications d'erreur et l'historique du projet.

### 2.3.3. Description de la barre d'outils :



	Créer un nouveau projet		Connecter un robot
	Ouvrir un projet		Déconnecter un robot
	Enregistrer le projet		Jouer une séquence
	Annuler l'action que tu viens de faire		Arrêter une séquence
	Rétablir l'action		Erreur



	Gérer le volume du robot		Passer en mode repos
	Passer en mode Animation		Activer/désactiver les moteurs
	Activer/désactiver la vie autonome		Niveau de batterie

## 2.4 Connexion au robot

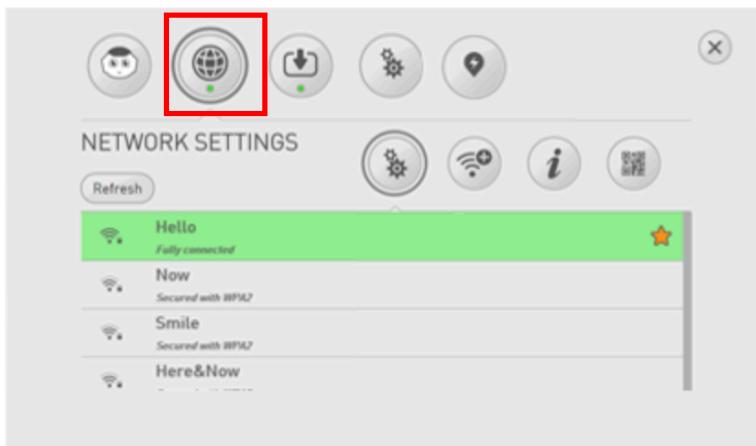
Pour utiliser votre robot réel sur Choregraphe, vous devez le connecter au même réseau que votre ordinateur.

### 2.4.1. Connexion WiFi

1. Accédez au menu des paramètres de la tablette de Pepper en cliquant sur



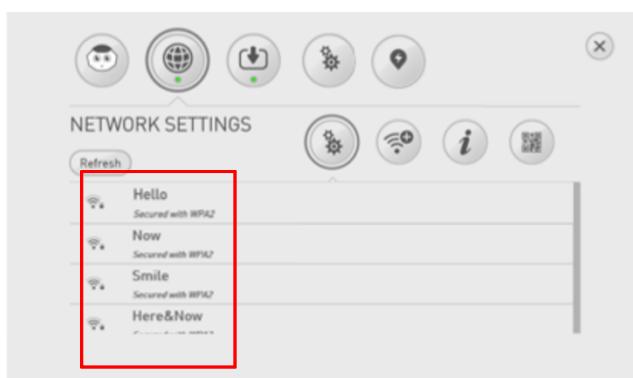
2. Allez sur l'onglet *Paramètres réseau* et sur la page *Connexion standard au WiFi*



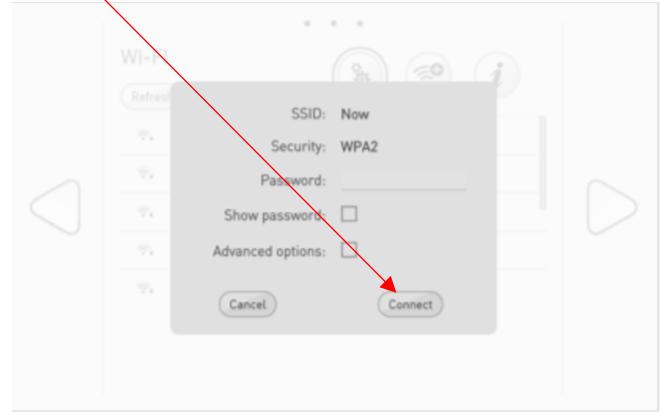
Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)

3. Sélectionnez un réseau WiFi

4. Complétez le mot de passe et cliquez sur le bouton Connect

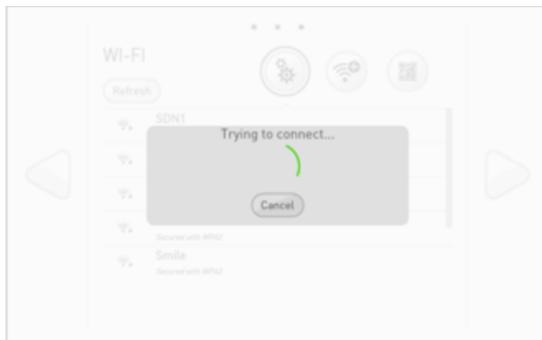


Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)



Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)

5. Attendez que la connexion se fasse. Une étoile apparaît à côté de votre connexion.

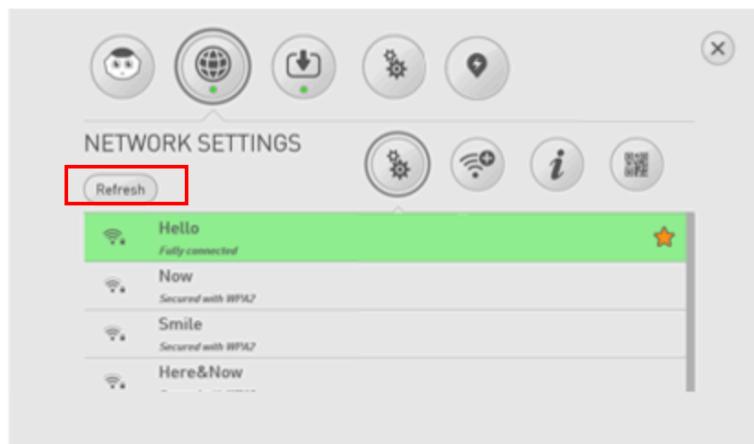


Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)



Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)

6. Si la connexion que vous souhaitez n'apparaît pas, rafraîchissez la page.

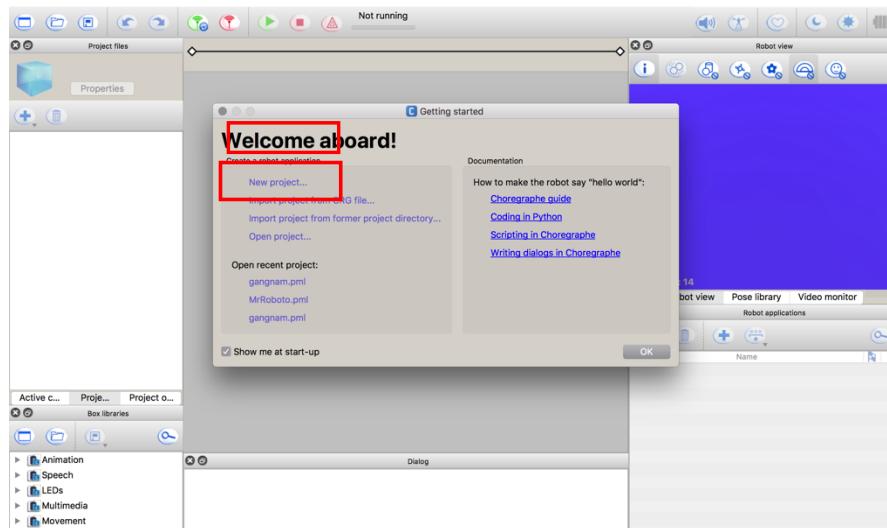


Softbank robotics documentation, [doc.aldebaran.com](http://doc.aldebaran.com)

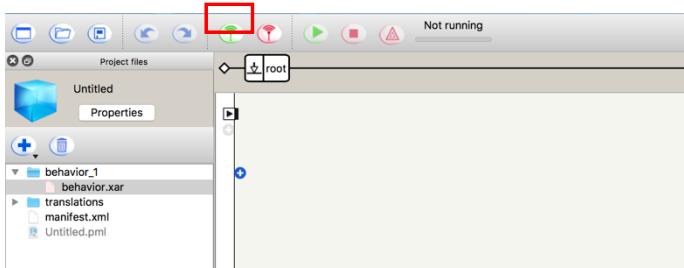
7. Connectez votre ordinateur au même réseau.



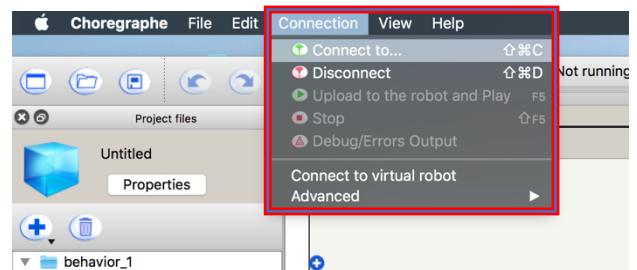
8. Démarrez le logiciel Choregraphe et choisissez *New project* ou *Open project*.



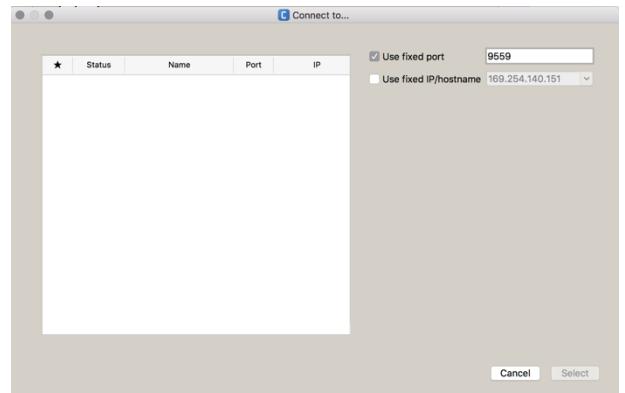
9. Cliquez sur le logo de connexion  ou sur l'onglet *Connexion – Connect To*.



ou



8. La fenêtre *Connect to* apparaît. Double-cliquez sur le robot de votre choix. L'image du robot doit être verte. S'il apparaît d'une autre couleur nous te conseillons de redémarrer Choregraphe pour éviter d'éventuels bogues.



#### 2.4.2. Utilisation d'un routeur WiFi multifonction

Il peut être pratique d'utiliser un routeur WiFi Multifonction pour connecter vos appareils entre eux. De cette manière, partout où vous allez, vous êtes assuré de pouvoir utiliser Choregraphe. Voici la façon d'utiliser notre borne :

1. Allumez le Tripmate Hoo Too (routeur Wi-Fi multifonction) en appuyant 2-3 secondes sur le bouton de démarrage. Les voyants clignotent à tour de rôle, puis le voyant bleu clignote avant d'être allumé en continu.



2. Connectez-la à Pepper via les paramètres réseaux de la tablette. Pour cela, repérez le dans les réseaux Wi-Fi. Le mot de passe est 11111111 (le chiffre 1 huit fois).

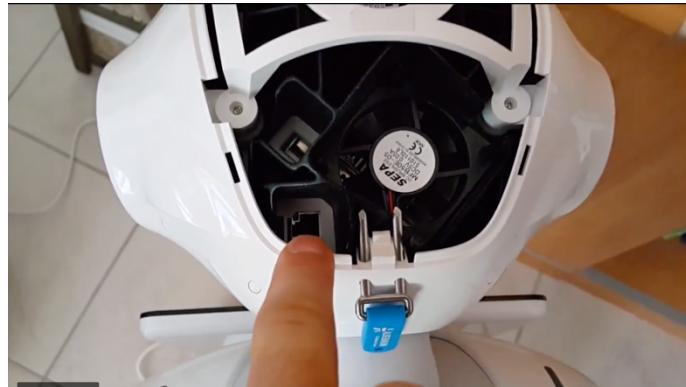
3. Connectez-la également à votre ordinateur.

### 2.4.3. Connexion Filaire

Si vous n'avez aucune façon de connecter votre robot par WiFi, vous pouvez le faire grâce à un câble Ethernet RJ45.



1. Prenez la clé de frein du bassin et insérez-là à l'arrière de la tête.
2. Enlevez le cache. Vous pouvez maintenant accéder au port Ethernet du robot.



3. Connectez votre câble au robot et à votre ordinateur. La connexion filaire a la priorité sur une connexion WiFi.
4. Allumez Choregraphe. Votre robot devrait apparaître dans la page *Connect to*.

### 2.4.4. Débogage de connexion

Pour chaque bogue essayez les étapes dans l'ordre chronologique, c'est-à-dire tentez la première solution, si celle-ci est défaillante, passez à la suivante.

Si Pepper apparaît d'une couleur autre que vert il est préférable de quitter Choregraphe et de le redémarrer.

Si Pepper n'apparaît pas dans le logiciel ou qu'il apparaît mais ne se connecte pas :

- Vérifiez que votre ordinateur soit bien connecté au même réseau WiFi que votrerobot, en comparant le réseau sélectionné dans les paramètres réseaux de la tablette de Pepper et celui sélectionné dans la liste de votre ordinateur.

- Vérifiez que la connexion soit satisfaisante en vous assurant que le logo Wi-Fi contient suffisamment de barres pour assurer la connexion.
- Redémarrez le logiciel.
- Vérifiez que l'adresse IP de Pepper indiqué sur le logiciel soit la même que celle qu'il énonce oralement. Pour cela, double-cliquez  sur le logo et identifiez l'adresse IP. Ensuite appuez sur le bouton central de votre robot pour qu'il énonce son adresse IP oralement. En appuyant à nouveau sur le bouton central, vous pouvez lui faire répéter. Si l'adresse IP est différente vous devez la changer manuellement sur le logiciel. Cochez la case *Use fixed IP/hostname*, entrez manuellement l'adresse IP de Pepper et cliquez sur *select*.
- Redémarrez le Tripmate.
- Redémarrez le robot.

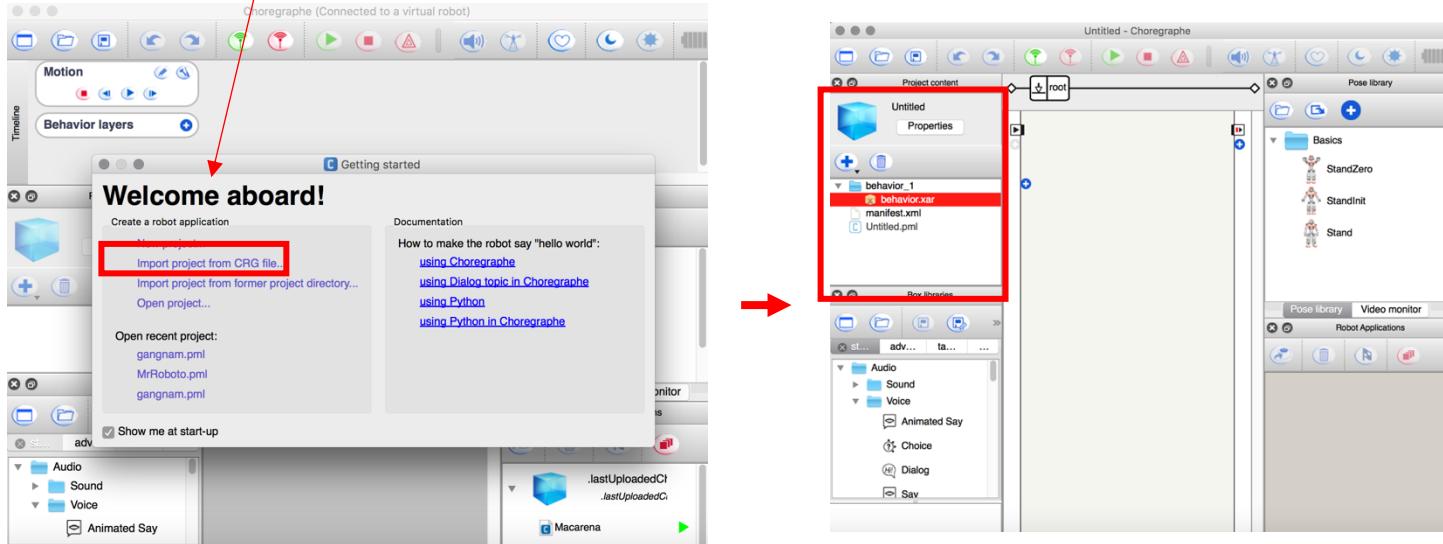
## 2.5 Crédation d'un nouveau projet



La Fenêtre *Getting started* apparaît.

1. Ouvrez le logiciel en double-cliquant sur son icône.

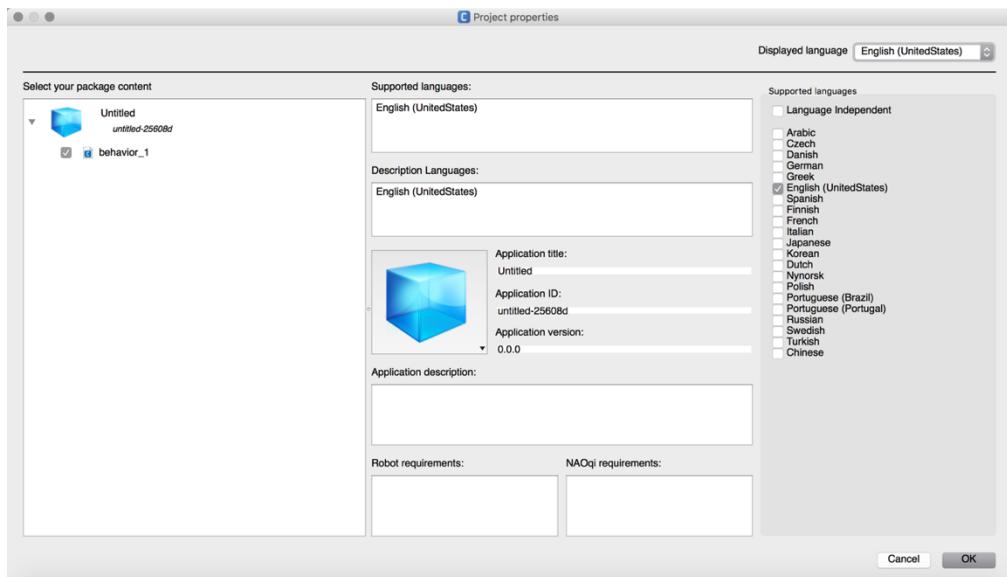
2. Cliquez sur New project. Une nouvelle fenêtre de travail s'ouvre.



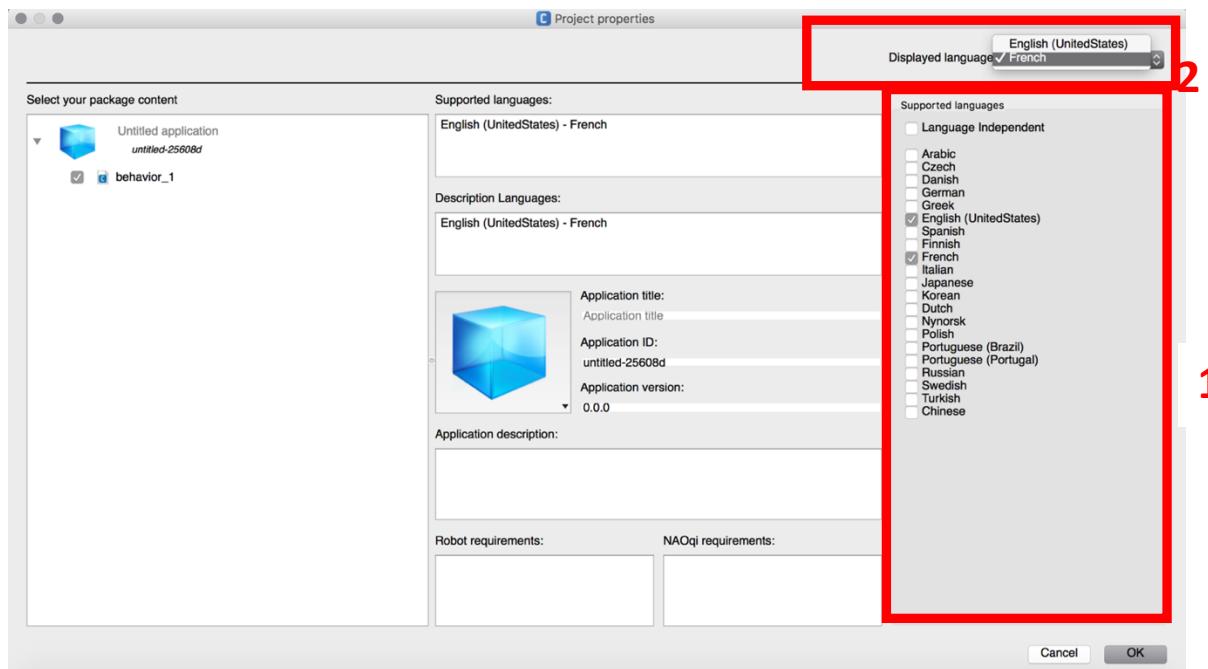


NB : Si Choregraphe est déjà ouvert, vous devez cliquer sur l'icône de votre barre d'outils, pour créer un nouveau projet. Un message d'enregistrement peut apparaître si vous travaillez déjà sur un projet.

3. Les informations de votre projet s'affichent dans la fenêtre *Project content*. Par défaut, le titre est *Untitled*. Pour définir de nouvelles propriétés, cliquez sur [Properties](#). La fenêtre *Project properties* apparaît.

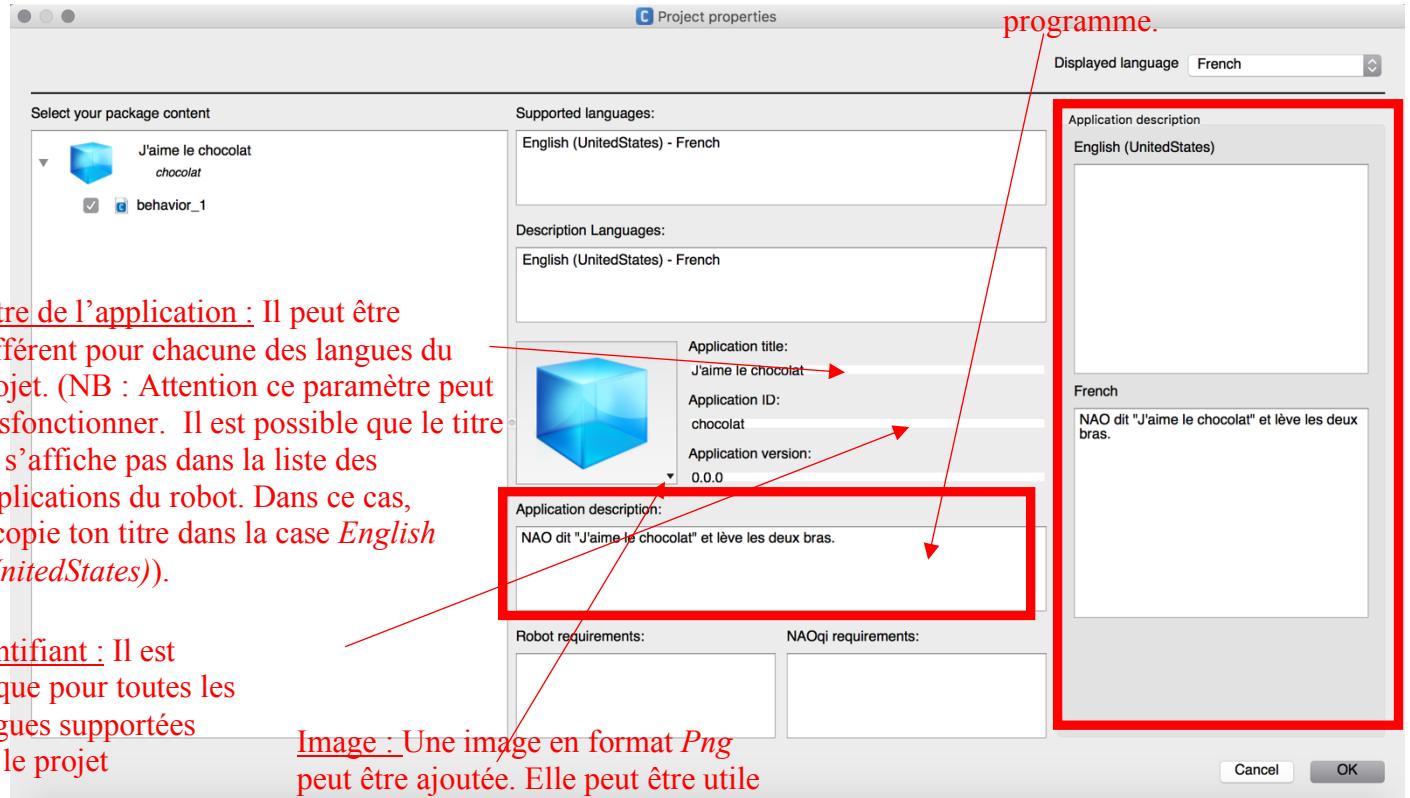


4. Définissez la langue de votre projet en cochant la case correspondante (1). Vous pouvez en choisir plusieurs. Puis choisissez la langue à afficher dans la fenêtre *Project properties* (2)



5. Donnez un nom à votre application et notez une description. Pour chacune de ces caractéristiques, cliquez sur le cadre correspondant pour l'éditer. Les détails de l'encadré sélectionné s'afficheront à droite de la fenêtre.

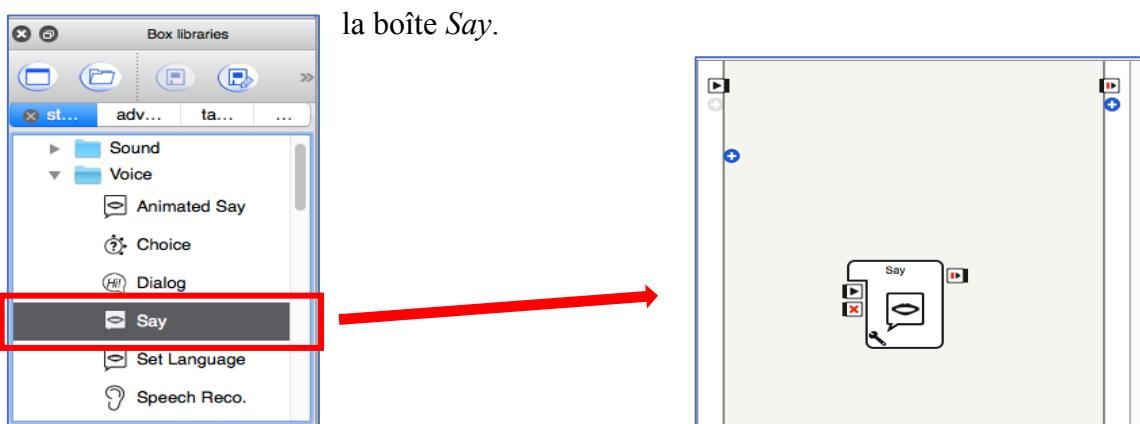
**Description:** Elle permet d'identifier très vite les actions du programme.



6. Cliquez sur *OK* pour revenir à ta fenêtre de travail. Vous pouvez maintenant programmer des comportements.

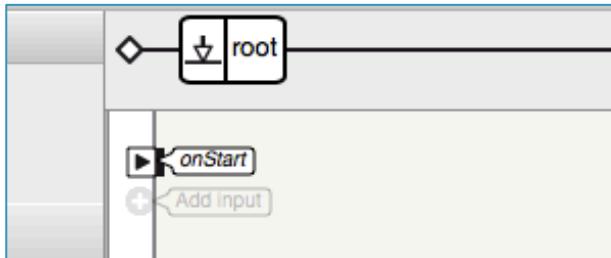
7. Pour commencer à manipuler le logiciel de programmation, servez-vous des boîtes de comportements préexistantes qui se trouvent dans la librairie. Elles sont diverses et variées. Glissez-les dans votre espace de travail. Par exemple, si vous souhaitez que Pepper parle, prenez

la boîte *Say*.

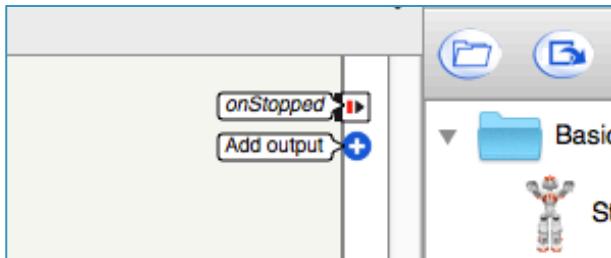


8. Pour que le programme fonctionne, il faut que vous connectez toutes les actions ensemble. On va appeler l'ensemble des actions « une *séquence* ».

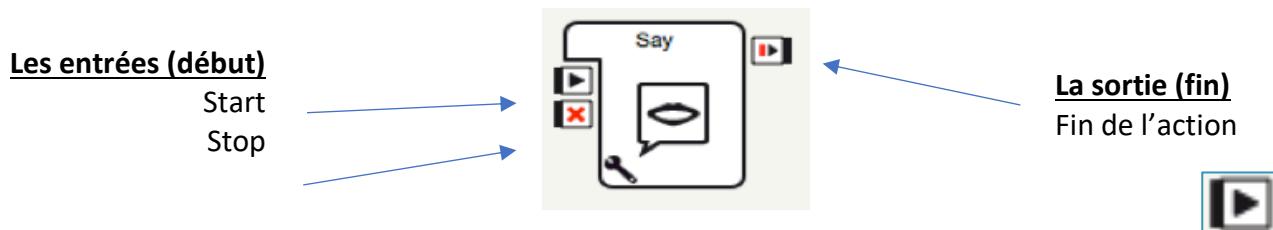
Dans votre espace de travail, vous voyez le **début de la séquence** (*onStart*) à gauche :



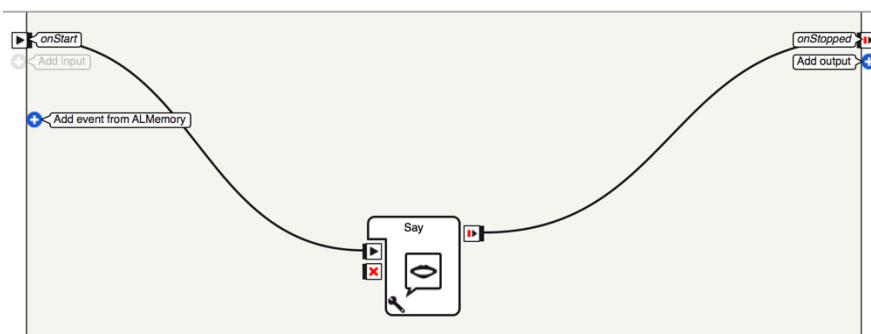
La **fin de la séquence** (*onStopped*) se trouve à droite :



Chaque boîte contient plusieurs connexions :



Pour que Pepper parle, vous devez connecter *onStart* à l'input *Start* de la boîte *Say* et l'output *Fin de l'action* à *onStopped*.



## 2.6. Les boîtes de comportements préexistantes

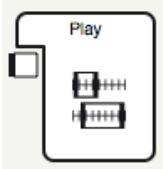
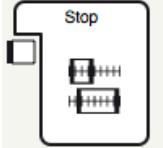
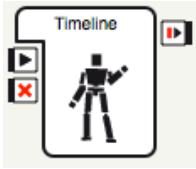
### 2.6.1. Animation



#### ▶ Creation

Ce dossier est composé de 5 boîtes qui peuvent être utiles lors de la création d'un comportement en mode *Timeline*.

	<b>Aller à (nom)</b>	Saute au nom de l'image ( <i>frame</i> ) programmé dans les paramètres.  N.B. L' <i>image</i> est un temps donné sur la ligne du temps (par ex. : l'image située à 20 s. sur la ligne du temps).
	<b>Aller à (nombre)</b>	Saute au nombre de l'image ( <i>frame</i> ) programmé dans les paramètres.

	<h2>Jouer</h2>	<p>Joue la ligne du temps parente depuis l'index d'image actuel. Ne fait rien si la ligne du temps est déjà en train de jouer.</p>
	<h2>Arrêt</h2>	<p>Arrête la ligne du temps parente à l'index d'image actuel.</p>
	<h2>Ligne du temps</h2>	<p>Cette boîte est vide et devrait être utilisée pour créer l'animation de votre choix. Double-cliquer sur la boîte pour la définir.</p>



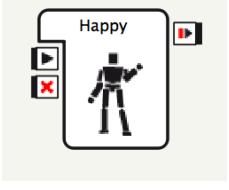
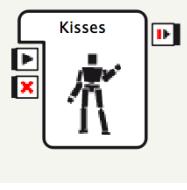
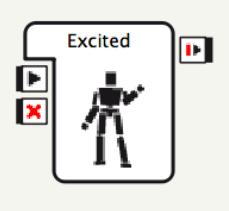
## Dialog animations

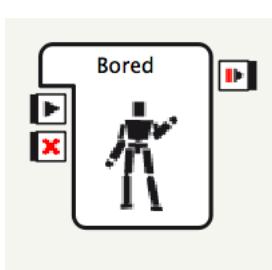
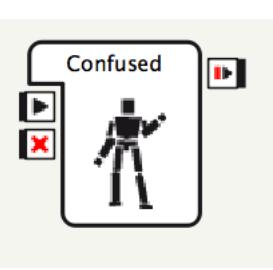
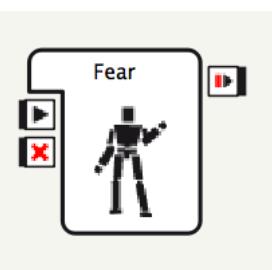
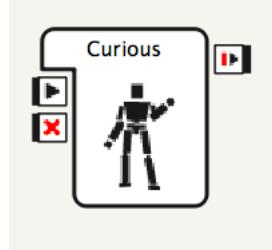
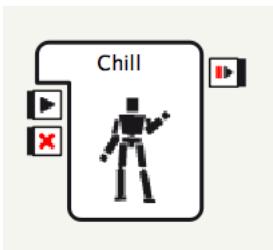
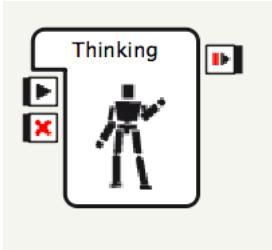
Ce dossier contient plus de 70 boîtes de mouvements à utiliser lors de discours ou de dialogues. Ils accompagnent différents types de phrases telles que affirmatives, exclamatives, interrogatives, etc...



## Moods

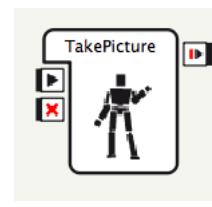
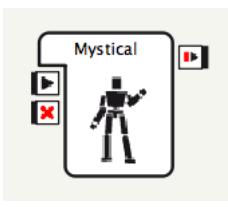
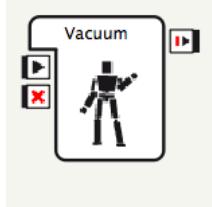
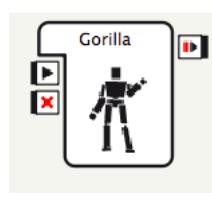
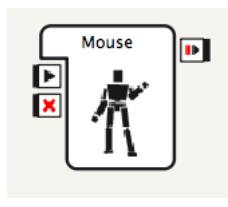
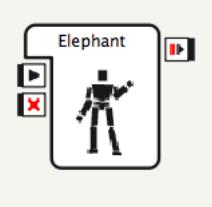
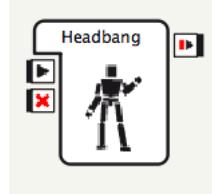
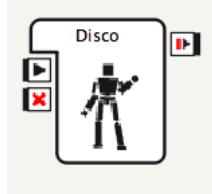
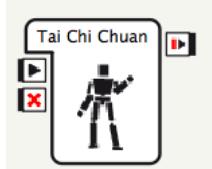
Ce dossier contient 9 boîtes qui expriment des humeurs différentes : positive, neutre ou négative.

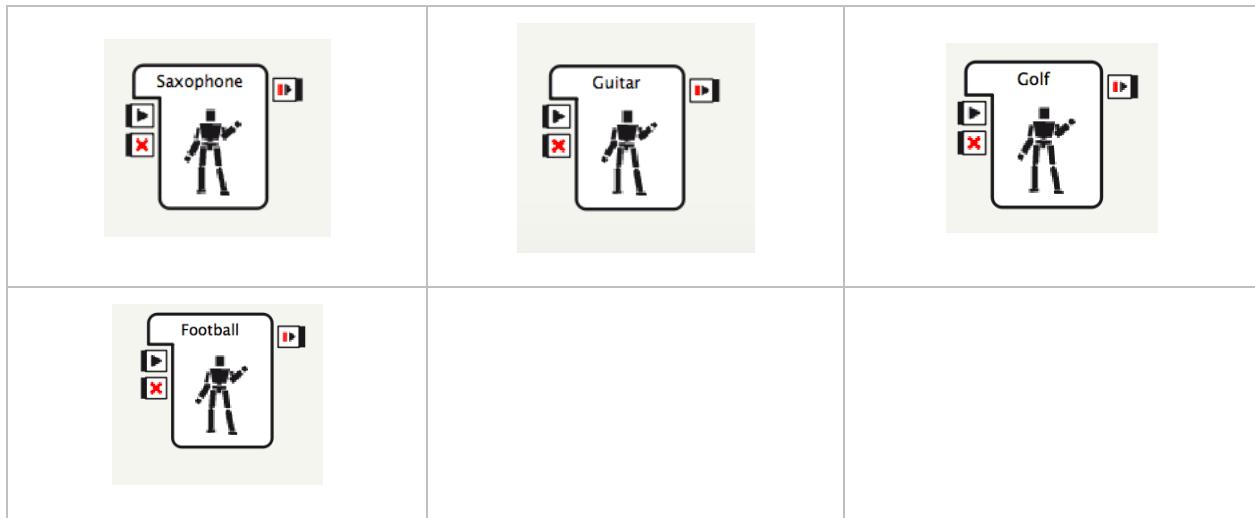
		
---	---	---



## Entertainment

Ce dossier contient 14 animations différentes : danses et mimes (instruments, animaux, sports...).





## 2.6.2 Speech (Discours)

- ▼ **Speech**
  - ▶ **Creation**
  - ▶ **Speech Settings**

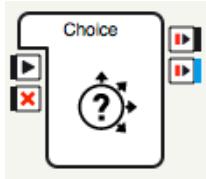
▼ **Creation**

- Animated Say**
- Choice**
- Choice (light)**
- Dialog**
- Say**
- Say Text**
- Speech Reco.**
- Animated Say Text**
- Get Localized Text**

▼ **Speech Settings**

- Set Language**
- Set Reco. Lang.**
- Set Speaker Vol.**
- Set Speech Lang.**

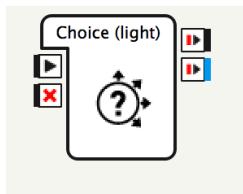
<p>Ce dossier contient 9 boîtes qui permettent de programmer un discours.</p>	<p><b>Discours animé (mouvements)</b></p> <p>Parle de façon animée. Vous devez ouvrir la boîte <i>Animated Say</i> pour entrer le texte à dire.</p>
---	---



## Choix de réponses

Le choix est basé sur la reconnaissance vocale : on s'attend à un certain nombre de réponses prédefinies.

La boîte inclus de l'aide interactive pendant le comportement : simplement appuyer sur les capteurs sensoriels de la tête.



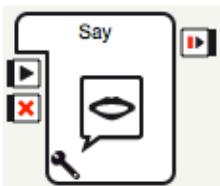
## Choix de réponse (simplifié)

Ceci est une version allégée de la boîte *Choice* de la bibliothèque standard qui n'a pas d'animation intégrée.



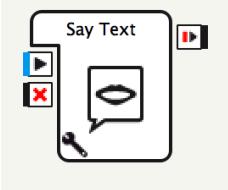
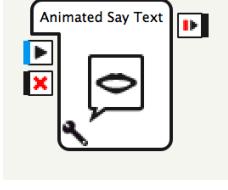
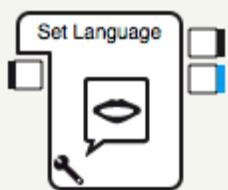
## Dialogue

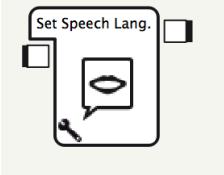
Un exemple de la mise en œuvre d'un dialogue plurilingue.



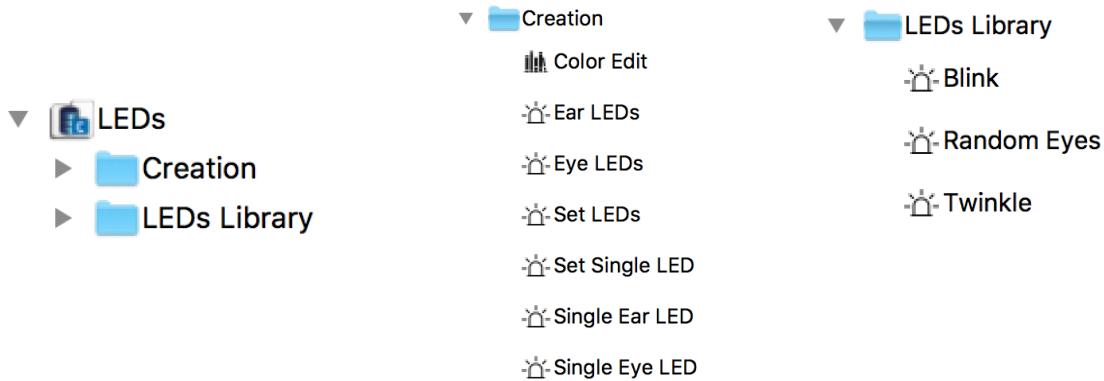
## Dire

Dis des lignes de textes écrites dans la boîte.

	<h3>Dire texte</h3>	<p>Dis le texte reçu dans l'entrée de la boîte.</p>
	<h3>Reconnaissance vocale</h3>	<p>Reconnait des mots à partir d'une liste écrite dans les paramètres de la boîte.</p>
	<h3>Dire du texte animé</h3>	<p>Dis le texte reçu dans l'entrée de la boîte de manière animée.</p>
	<h3>Récupérer un texte localisé</h3>	<p>Récupère un texte localisé dans les paramètres et l'envoyer à la sortie de la boîte.</p>
<p>▼  <b>Speech Settings</b></p> <p>Ce dossier contient 4 boîtes qui permettent de paramétrier la parole du robot.</p>		
	<h3>Programmer la langue</h3>	<p>Permet de sélectionner la langue qui sera comprise et parlée par PEPPER. Toutes les boîtes suivantes relatives au langage vont utiliser les langues programmées ici.</p>

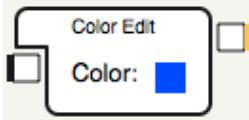
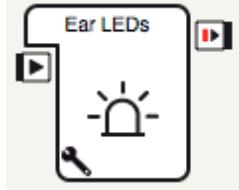
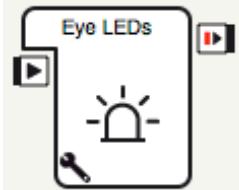
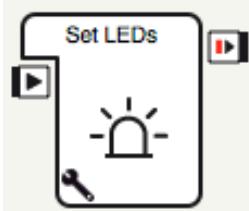
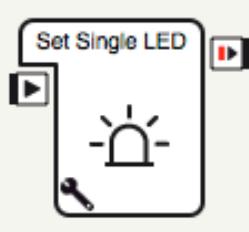
	<h2>Programmer la langue de la reconnaissance vocale.</h2>	<p>Permet de sélectionner la langue que vous souhaitez que le robot reconnaisse.</p>
	<h2>Ajuster le volume des haut-parleurs</h2>	<p>Définir le volume du son entre 0 (sourdine) et 100 (maximum).</p>
	<h2>Programmer la langue que le robot parle</h2>	<p>Permet de sélectionner la langue que vous souhaitez que le robot parle.</p>

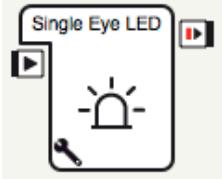
### 2.6.3 LEDs (Diodes électroluminescentes)



▼  **Creation**

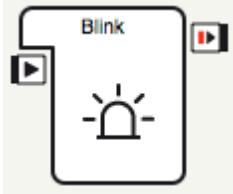
Ce dossier contient 7 boîtes qui permettent de programmer les yeux de PEPPER.

	<b>Définir une couleur</b>	Envoie une couleur encodée en valeurs de RVB.
	<b>DEL d'oreille</b>	Définit l'intensité de toutes les DEL d'une ou des deux oreilles.
	<b>DEL d'oeil</b>	Définit la couleur des yeux. Vous devez ouvrir la boîte pour définir la couleur.
	<b>Programmer les DEL</b>	Définit l'intensité des DEL d'un groupe choisi (dans la paramètres).
	<b>Programmer une seule DEL</b>	Définit l'intensité d'une DEL en particulier.

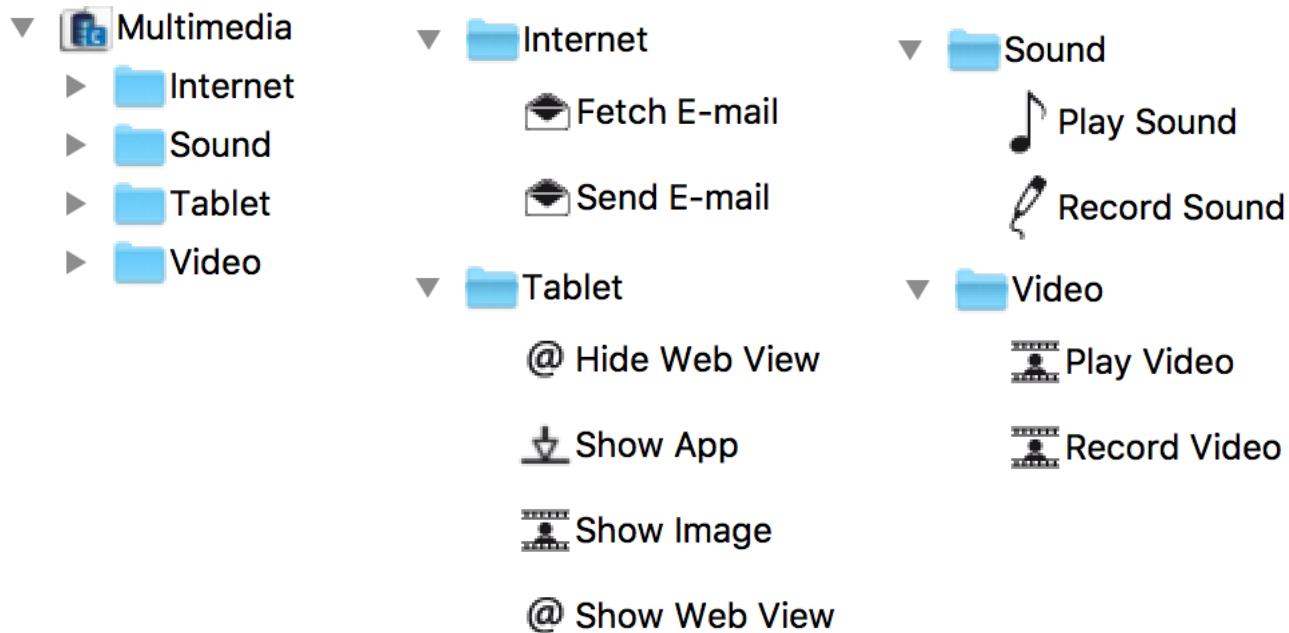
	<b>Un seul DEL d'oreille</b>	Définit l'intensité d'une DEL d'oreille en particulier.
	<b>Un seul DEL d'oeil</b>	Définit la couleur d'une DEL d'oeil en particulier.  Vous devez ouvrir la boîte pour définir la couleur.

▼  **LEDs Library**

Ce dossier contient 3 boîtes qui permettent de choisir la couleur des yeux de PEPPER.

	<b>Clignotement</b>	Fait clignoter les yeux de PEPPER une seule fois.
	<b>Yeux aléatoires</b>	Fait changer la couleur des yeux de façon aléatoire.  Note : c'est une boîte qui ne s'arrête pas. Vous devez stimuler l' <i>input onStop</i> pour l'arrêter.
	<b>Clignotement continu</b>	Les DEL sélectionnées clignotent pendant une durée définie dans les paramètres.

## 2.6.4 Multimedia



### ► Internet

Ce dossier contient 2 boîtes qui permettent de recevoir ou d'envoyer des courriels.

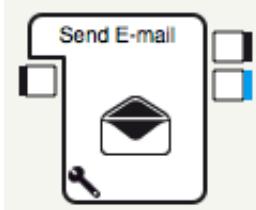


### Recevoir un courriel

Permet de recevoir des courriels à partir d'un serveur, qui doit être programmé dans les paramètres.

Note : s'assurer que POP est activé sur le serveur.

**Attention** : le mot de passe doit être entré dans le comportement et n'est pas encrypté, ni protégé. Il est donc préférable de créer une adresse courriel distincte pour le robot.

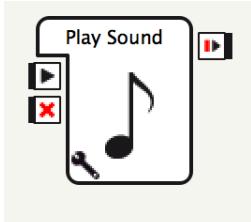


## Envoyer un courriel

Permet d'envoyer des courriels.  
Définir tous les paramètres de ce courriel avant de l'envoyer.

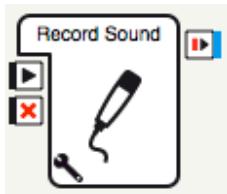
### ▼ Sound

Ce dossier contient 2 boîtes qui permettent de programmer du son.



## Jouer un son

Lis un fichier son. Sélectionnez un fichier dans les paramètres en format mp3.



## Enregistrer un son

Permet d'enregistrer un son sur le robot à partir du microphone choisi.  
L'*output* est stimulé à la fin de l'enregistrement.

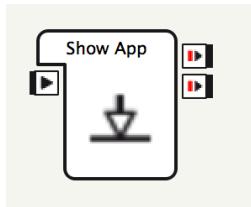
### ► Tablet

Ce dossier contient 4 boîtes qui permettent de programmer la tablette



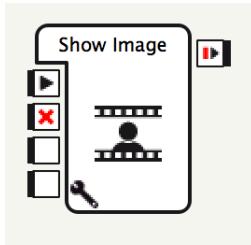
## Masquer la vue Web

Masque le contenu html derrière une couverture générique.



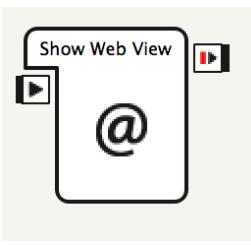
## Afficher l'application

Charge l'application sur la tablette, si elle existe, et affiche la vue Web.



## Afficher une image

Affiche une image à l'écran

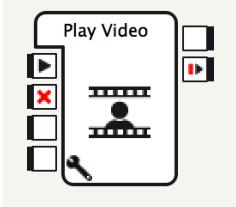


## Afficher la vue Web

Affiche le contenu html sur la tablette.

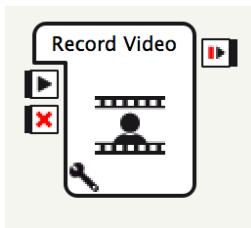


Ce dossier contient 2 boîtes qui permettent de programmer des vidéos sur la tablette.



## Jouer une vidéo

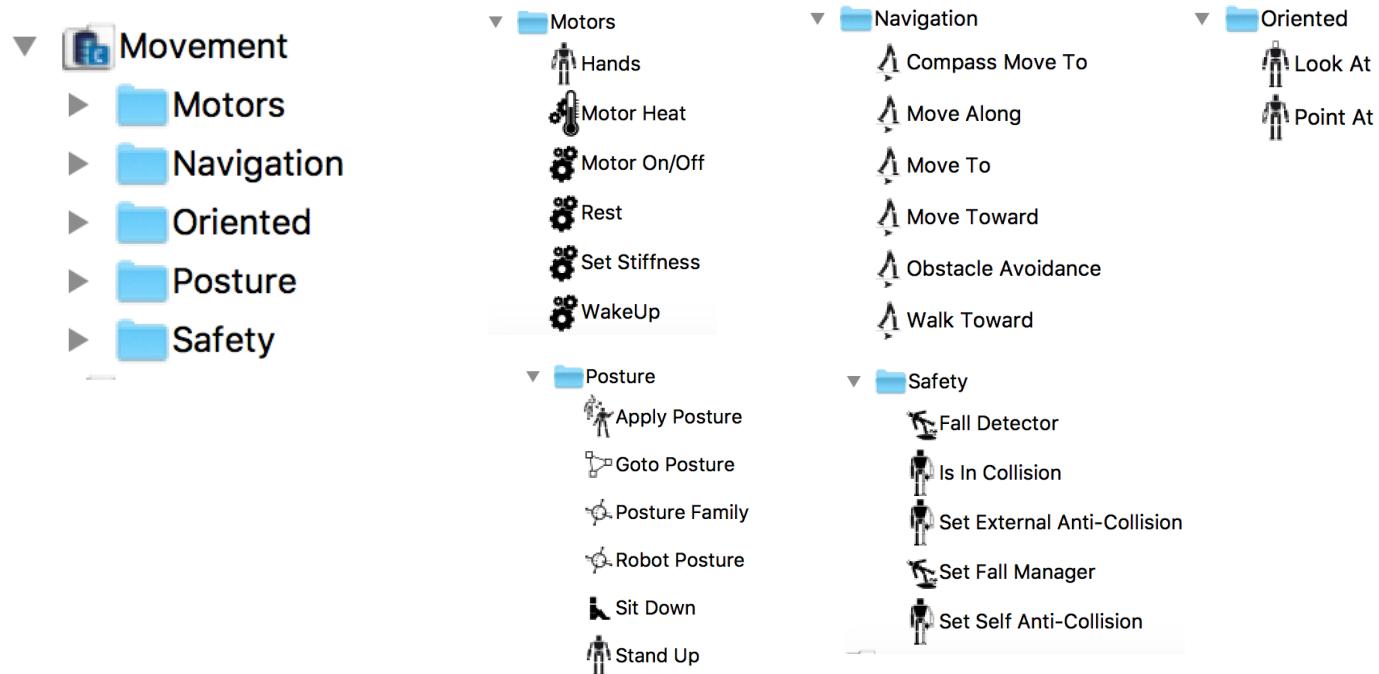
Affiche une vidéo sur la tablette en format mp4.



## Enregistrer une vidéo

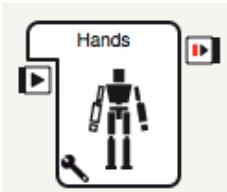
Enregistre une vidéo avec une des caméras du robot. Arrête l'enregistrement lorsque la durée définie dans les paramètres est écoulée. Par défaut, le fichier sera stockée sur le robot dans /home/nao/ recording / camera

## 2.6.5 Movement (Mouvements)



### ► Motors

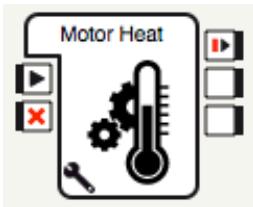
Ce dossier contient 6 boîtes qui permettent de programmer les moteurs du robot.



#### Mains

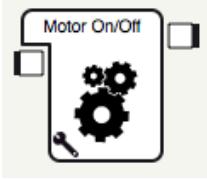
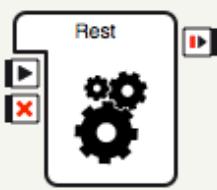
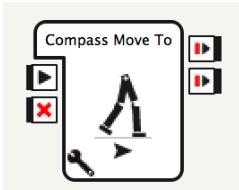
PEPPER raidit les moteurs d'une ou de ses mains pour qu'il puisse les fermer/ouvrir.

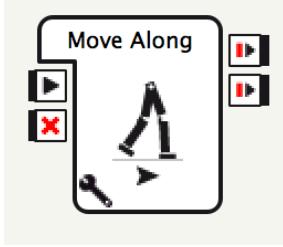
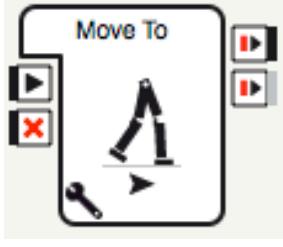
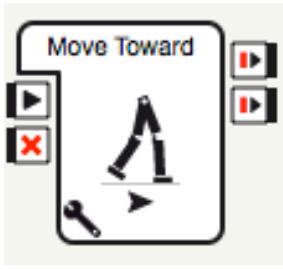
Il détend ensuite les moteurs de ses mains.



#### Température du moteur

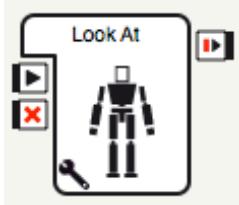
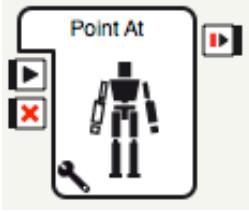
Vérifie la température du moteur ou d'un groupe de moteurs et notifie lorsqu'il devient chaud ou froid.

	<h2>Moteurs</h2> <h3>Marche/Arrêt</h3>	<p>Raidit ou détend tous les moteurs.</p>
	<h2>Repos</h2>	<p>Appelle un processus de repos : place le robot dans une position accroupie puis détend toutes les articulations.</p>
	<h2>Régler la rigidité</h2>	<p>Renforce les moteurs sélectionnés dans les paramètres.</p>
	<h2>Réveil</h2>	<p>Appelle un processus de réveil : raidit toutes les articulations et place le robot dans une position debout (<i>stand Init</i>).</p>
<span data-bbox="225 1227 262 1262">▶</span>  <b>Navigation</b>		
	<h2>Se déplacer vers un point configuré</h2>	<p>Déplace le robot vers un point configuré par rapport à son emplacement actuel en utilisant la caméra pour corriger la déviation.</p>

	<h2>Se déplacer sur une trajectoire.</h2>	<p>Se déplace sur une trajectoire donnée par un fichier attaché .pmt.</p>
	<h2>Aller à</h2>	<p>Fait marcher le robot vers un point configuré relativement à la position actuelle.</p>
	<h2>Aller vers...</h2>	<p>Fait marcher le robot dans une direction définie dans les paramètres. Attention : le robot ne va pas arrêter de marcher par lui-même. Il faut corriger sa trajectoire ou arrêter la boîte pour qu'il s'arrête.</p>
	<h2>Éviter les obstacles</h2>	<p>Marche vers l'avant et tourne à droite lorsqu'il y a un obstacle.</p>
	<h2>Marcher vers</h2>	<p>Fait marcher le robot dans la direction qui a été définie dans les paramètres.</p>

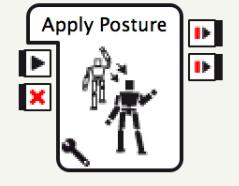
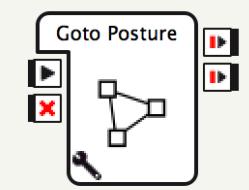
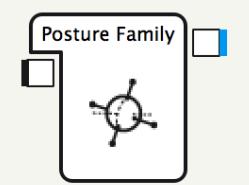
►  Oriented

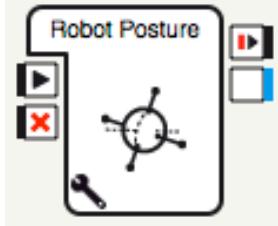
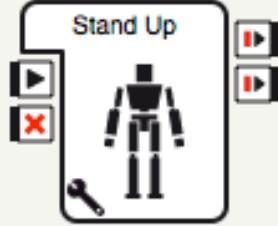
Ce dossier contient 2 boîtes qui permettent de programmer l'orientation du robot.

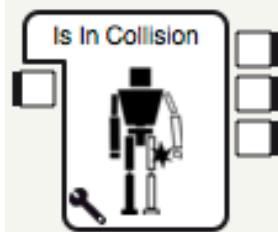
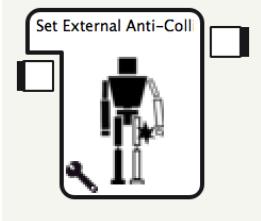
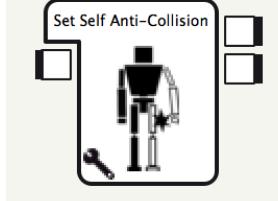
	<h2>Regarder vers...</h2>	<p>Fait regarder le robot vers l'endroit désiré.</p>
	<h2>Pointer vers...</h2>	<p>Fait pointer le robot vers l'endroit désiré.</p>

►  Posture

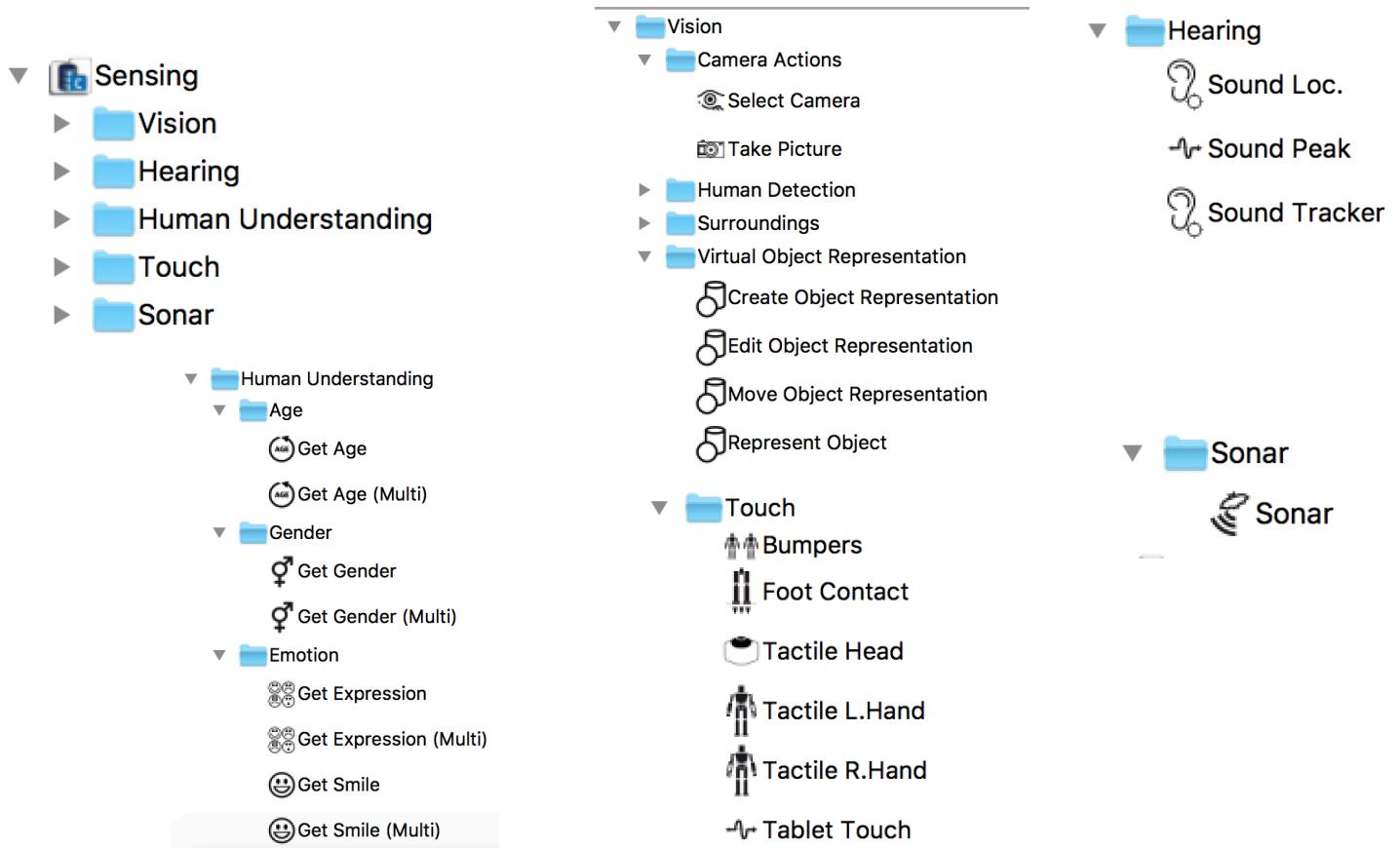
Ce dossier contient 6 boîtes qui permettent de configurer des postures.

	<h2>Appliquer la posture</h2>	<p>Définit directement toutes les articulations du robot dans la posture demandée.</p>
	<h2>Aller à la posture</h2>	<p>Le robot passe de sa position actuelle à la posture demandée.</p>
	<h2>Famille de posture</h2>	<p>Renvoie à une famille de posture.</p>

	<h2>Posture du robot</h2>	<p>Renvoie le nom de la posture du robot chaque fois que la valeur change.</p>
	<h2>S'asseoir</h2>	<p>Le robot tente de s'asseoir (nombre défini d'essais) à partir de n'importe quelle position. Vous pouvez définir le nombre d'essais dans les paramètres.</p>
	<h2>Se lever</h2>	<p>Le robot tente de se lever (nombre défini d'essais) à partir de n'importe quelle position. Vous pouvez définir le nombre d'essais dans les paramètres.</p>
<p>►  Safety</p> <p>Ce dossier contient 5 boîtes qui permettent de contrôler et de préserver le bon état du robot.</p>		
	<h2>Détecteur de chute</h2>	<p>Déetecte si le robot est tombé et s'il a activé le processus de gestion de chute.</p>

	<p><b>Est-ce qu'il y a une collision ?</b></p>	<p>Vérifie si la partie du corps sélectionnée dans les paramètres est en collision avec une autre partie du corps.</p>
	<p><b>Paramétrer la protection anticollision</b></p>	<p>Active ou désactive la protection anticollision externe sur une partie spécifique du corps du robot.</p>
	<p><b>Paramétrer le gestionnaire d'équilibre</b></p>	<p>Active ou désactive le gestionnaire d'équilibre</p>
	<p><b>Paramétrer la protection contre les collisions.</b></p>	<p>Active ou désactive la protection contre les collisions sur une partie spécifique du corps du robot.</p>

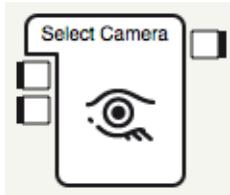
## 2.6.6 Sensing (Détection)



### Vision

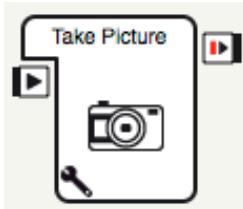
Ce dossier contient 23 boîtes, réparties dans 4 sous-dossiers : *Camera actions* permet de choisir les caméras et de prendre des photos, *Human Detection* permet d'apprendre au robot à reconnaître des personnes, *Surroundings* permet de prendre contact avec les alentours et *Virtual object representation* permet de configurer la représentation d'objets virtuels.

### Camera Actions



## Sélectionner une caméra

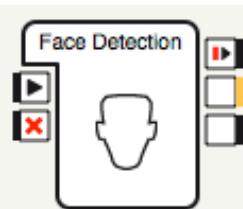
Sélectionne la caméra à utiliser.



## Prendre une photo

Prend une photographie à l'aide d'une des caméras.

### Human Detection



## Détection faciale

Détecte le visage de gens et renvoie le nombre de visages détectés.



## Reconnaissance faciale

Détecte le visage des gens et reconnaît ceux qui sont connus du robot (qui ont été appris, voir *Apprendre un visage*).



## Détecteur de visage

Fait détecter et suivre les visages avec différents modes.



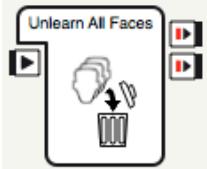
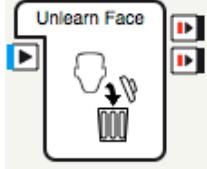
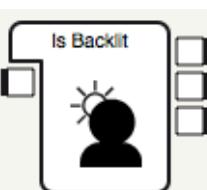
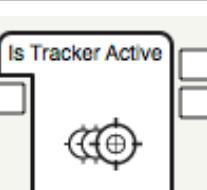
## Apprendre un visage

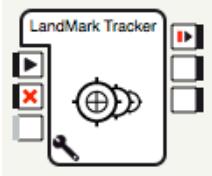
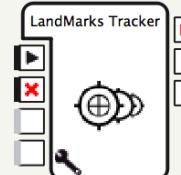
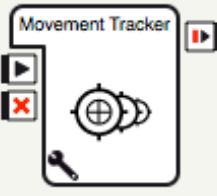
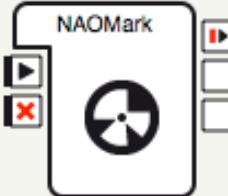
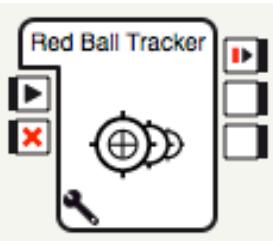
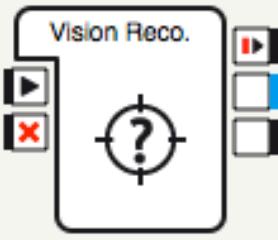
Fait apprendre au robot un nouveau visage à reconnaître. Les yeux du robot deviennent verts en cas de succès, et rouges en cas d'échec.



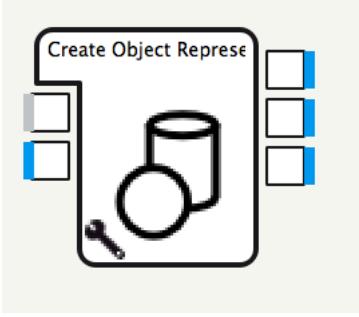
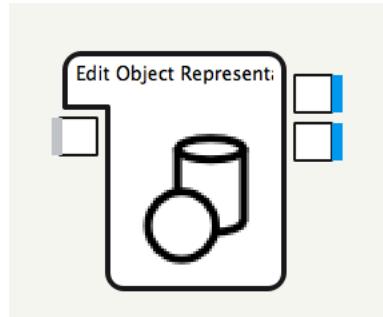
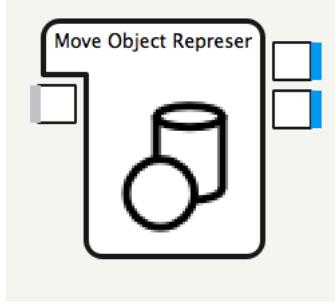
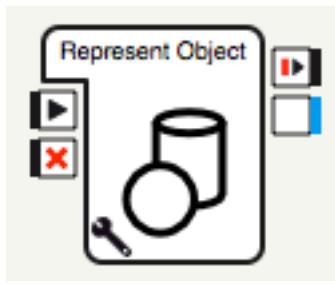
## Détecteur de personnes

Fait détecter et suivre les gens avec différents modes.

	<h2>Oublier tous les visages</h2>	<p>Supprime tous les visages appris et enregistrés dans la banque de données.</p>
	<h2>Oublier des visages</h2>	<p>Supprimer les visages associés au(x) nom(s) défini(s) dans l'entrée.</p>
<span style="color: #4f81bd; font-size: 1.5em;">📁</span> Surroundings		
	<h2>Conscience de base</h2>	<p>Cette boîte est une façon simple d'établir et de maintenir un contact visuel avec les gens.</p>
	<h2>L'image est-elle en contre-jour ?</h2>	<p>Vérifie si l'image des caméras est en contre-jour.</p>
	<h2>Suis-je dans l'obscurité ?</h2>	<p>Vérifie, en utilisant les caméras, si le robot est dans l'obscurité.</p>
	<h2>Le détecteur est-il actif ?</h2>	<p>Vérifie sur le détecteur est actif. La sortie <i>Yes</i> est stimulé si le détecteur est actif, et <i>No</i> s'il ne l'est pas.</p>

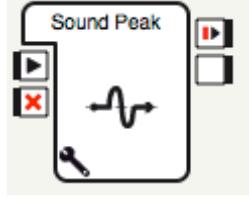
	<h2>Détecteur de point de repère</h2>	<p>Fait détecter et suivre un point de repère avec différents modes.</p>
	<h2>Détecteur de points de repère</h2>	<p>Fait détecter et suivre un motif avec deux points de repère.</p>
	<h2>Détecteur de mouvement</h2>	<p>Fait tourner la tête du robot dans la direction où il y a le plus de mouvements.</p>
	<h2>«NAO marques»</h2>	<p>Déetecte les <i>NAO marques</i> et renvoie le numéro d'identification de celles-ci.</p>
	<h2>Détecteur de balle rouge</h2>	<p>Fait détecter et suivre une balle rouge avec différents modes.</p>
	<h2>Reconnaissance visuelle</h2>	<p>Reconnait des images, des objets ou des lieux qui sont connus du robot (qui ont été appris).</p>

## Virtual Object Representation

	<h3>Créer la représentation du monde du robot</h3>	Permet d'ajouter ou d'enlever des objets de la représentation du monde du robot.
	<h3>Editer la représentation du monde du robot</h3>	Permet de modifier les propriétés d'affichage des objets de la représentation du monde du robot.
	<h3>Déplacer des objets</h3>	Permet de déplacer des objets de la représentation du monde du robot.
	<h3>Ajouter ou enlever un objet</h3>	Permet d'ajouter ou d'enlever un objet de la représentation du monde du robot.

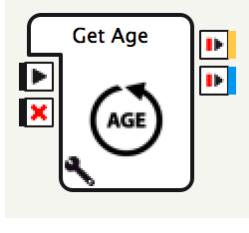
## Hearing

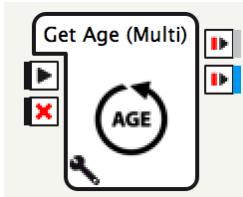
Ce dossier contient 3 boîtes qui permettent de détecter les sons.

	<b>Localisation du son</b>	Chaque fois qu'un son est détecté, la boîte envoie 2 <i>outputs</i> : (1) la localisation de la source du son par rapport à la tête du robot et (2) la position de la tête du robot par rapport au reste de son corps.
	<b>Détecteur de pic sonore</b>	Détecte les pics sonores. Chaque fois qu'il y a un pic sonore, l' <i>output</i> est stimulé. Vous pouvez ajuster la sensibilité de détection dans les paramètres.
	<b>Détecteur de son</b>	Fait détecter et suivre un son avec différents modes.

## Human Understanding

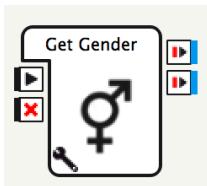
Ce dossier contient 10 boîtes, classées en 3 sous-catégories : *Age* permet de définir l'âge de l'humain, *Gender* permet de reconnaître le genre de l'humain et *Emotion* permet de détecter différentes émotions

	<b>Age</b>	Renvoie l'âge de la personne devant le robot. Si le robot détecte plusieurs personnes, le programme échoue.
---	------------	---



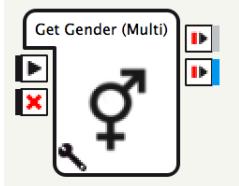
## Déetecter les âges

Renvoie l'âge de chaque personne devant le robot.



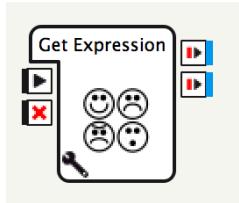
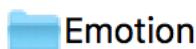
## DéTECTER LE GENRE

Renvoie le genre de la personne devant le robot. Si le robot détecte plusieurs personnes, le programme échoue.



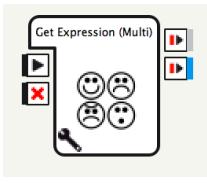
## DéTECTER LES GENRES

Renvoie le genre de chaque personne devant le robot



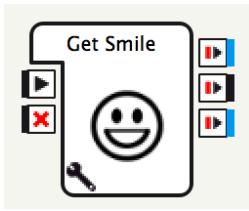
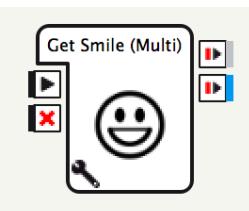
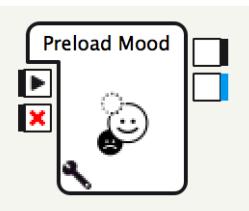
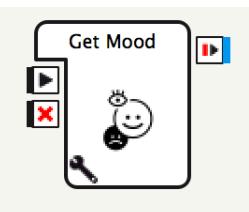
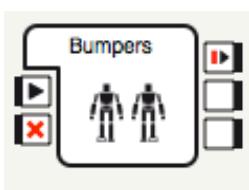
## DéTECTE L'EXPRESSION

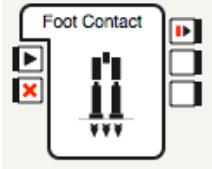
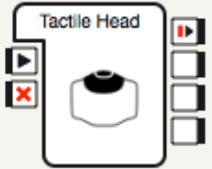
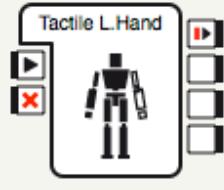
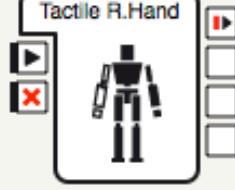
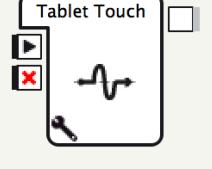
Renvoie l'expression faciale de la personne devant le robot. Si le robot détecte plusieurs personnes, le programme échoue.



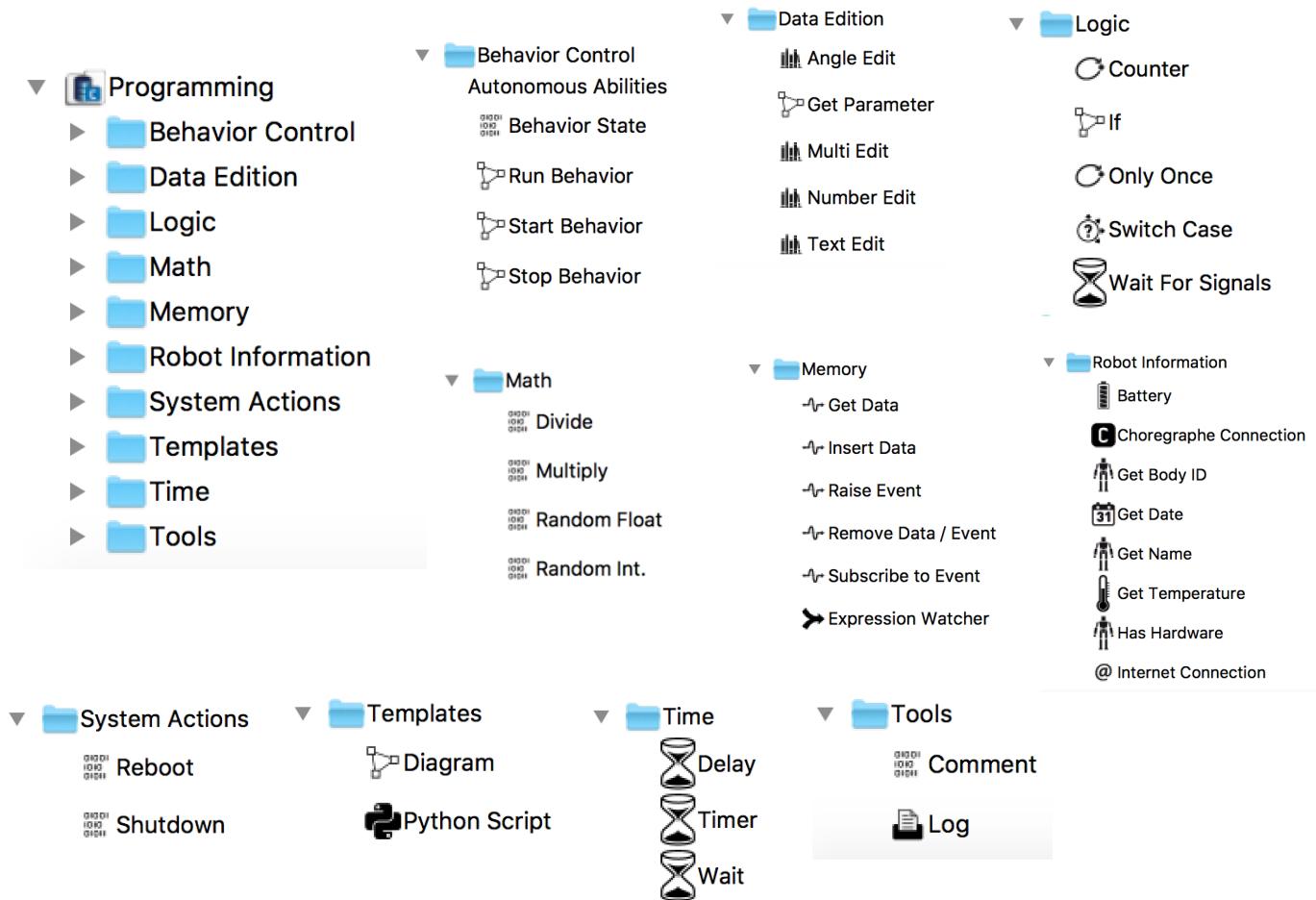
## DéTECTE LES EXPRESSIONS

Renvoie l'expression faciale de chaque personne devant le robot

	<h2>Déetecte le sourire</h2>	<p>Retourne si la personne devant le robot sourit ou non. Si le robot détecte plusieurs personnes, le programme échoue.</p>
	<h2>Déetecte les sourires</h2>	<p>Retourne si chaque personne devant le robot sourit ou non.</p>
	<h2>Initialiser l'humeur</h2>	<p>Initialise l'humeur du robot</p>
	<h2>Retourner l'humeur</h2>	<p>Retourne la réaction émotionnelle de l'utilisateur cible.</p>
<h3>Touch</h3>		
<p>Ce dossier contient 6 boîtes qui permettent de programmer les différents capteurs du robot.</p>		<p>Vérifie les détecteurs des amortisseurs. Stimule l'<i>output onLeft</i> (gauche) ou <i>onRight</i> (droite) en fonction de l'amortisseur qui est stimulé.</p>

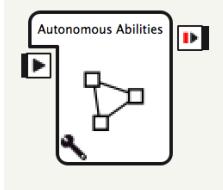
	<h2>Contact des pieds</h2>	<p>Déetecte des changements dans le contact des pieds avec le sol.</p>
	<h2>Détecteur tactile de la tête</h2>	<p>Déetecte un toucher sur les détecteurs tactiles de la tête.</p>
	<h2>Détecteur tactile de la main gauche</h2>	<p>Déetecte un toucher sur les détecteurs tactiles de la main gauche.</p>
	<h2>Détecteur tactile de la main droite</h2>	<p>Déetecte un toucher sur les détecteurs tactiles de la main droite.</p>
	<h2>Détecteur tactile de la tablette</h2>	<p>Déetecte un toucher sur la tablette</p>
 <b>Sonar</b>	<p>Ce dossier contient une boîte qui permet de détecter les obstacles.</p>	
	<h2>Sonar</h2>	<p>Déetecte les obstacles à l'aide de détecteurs ultrasoniques.</p>

## 2.6.7 Programming (Programmation)



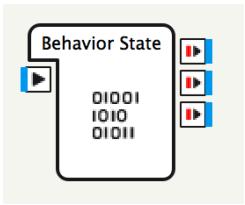
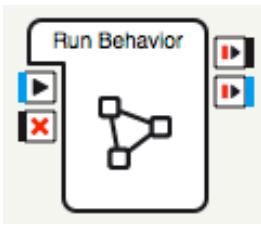
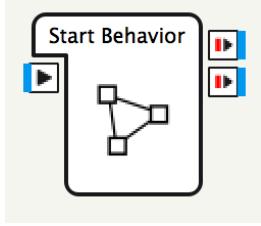
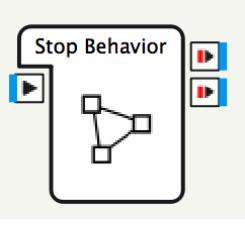
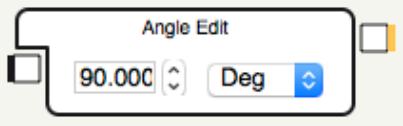
### Behavior Control

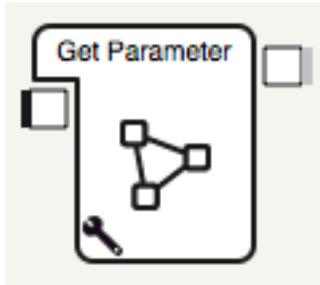
Ce dossier contient 5 boîtes qui permettent de contrôler les comportements du robot.



### Capacités autonomes

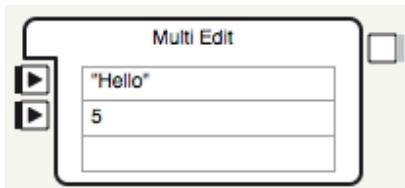
Gère les capacités autonomes du robot.

	<h2>État du comportement</h2>	<p>Retour sur différentes sorties en fonction de l'état du comportement.</p>
	<h2>Démarrer un comportement</h2>	<p>Permet de gérer un comportement. Débute le comportement défini en tant qu'<i>input</i> lorsqu'<i>onStart</i> est stimulé, et l'arrête lorsque <i>onStop</i> est stimulé.</p>
	<h2>Démarrer un comportement</h2>	<p>Démarre le nom du comportement donné en entrée.</p>
	<h2>Arrêter un comportement</h2>	<p>Arrête le comportement donné en entrée.</p>
 <b>Data Edition</b>	<p>Ce dossier contient 5 boîtes qui permettent d'éditer les données.</p>	
	<h2>Définir un angle</h2>	<p>Envoie un angle. L'angle est converti en radians, puis est envoyé à travers la sortie.</p>



## Obtenir le paramètre

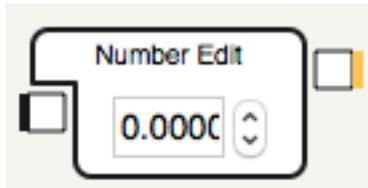
Retourne la valeur du paramètre de la boîte parente. Établir le nom du paramètre qui devra être retourné dans les paramètres de cette boîte.



## Définir plusieurs valeurs

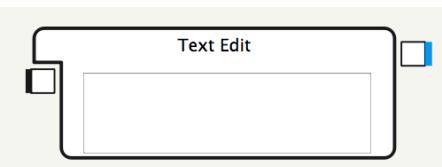
Envoie la valeur qui se trouve devant l'*input* stimulé (peut être tous les types de langages de Python : string, int, float, array, etc.).

Vous pouvez définir une valeur en double-cliquant sur la ligne. Pour insérer une ligne : cliquez droit + *insert row*.



## Définir un nombre

Envoie le nombre défini lorsque l'*input* est stimulé.

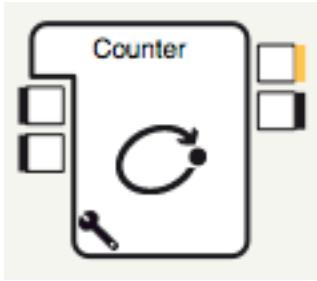
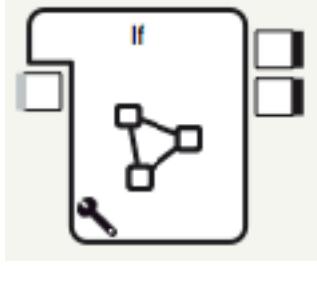


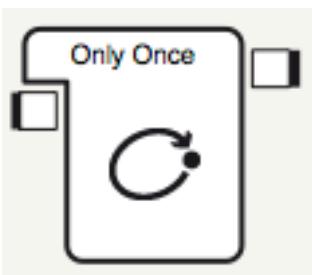
## Définir un texte

Envoie le texte qui a été défini à la stimulation de l'entrée

## Logic

Ce dossier contient 5 boîtes qui permettent de définir des conditions.

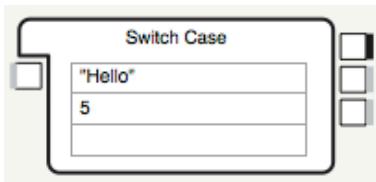
	<h2>Compteur</h2>	<p>Représente un compteur.</p> <p>Chaque fois que l'<i>input onStart</i> est stimulé, la valeur du compteur est envoyée à l'<i>output currentValue</i>, et incrémentée (ou décrémentée) par <i>Step value</i>, de sa valeur initiale à sa valeur finale programmée dans les paramètres. Lorsque la valeur du compteur est supérieure à la valeur finale, l'<i>output onReinitialized</i> est stimulé et le compteur est réinitialisé à sa valeur initiale.</p> <p>Note : vous pouvez aussi réinitialiser le compteur en stimulant l'<i>input onInit</i>.</p>
	<h2>Si...</h2>	<p>Teste une condition et stimule l'<i>output</i> approprié en fonction de la valeur booléenne (vrai ou faux) de la condition.</p> <p>La condition peut être programmée dans les paramètres.</p>



## Seulement une fois

L'*output* de cette boîte est stimulé seulement la première fois que l'*input onSignal* est envoyé.

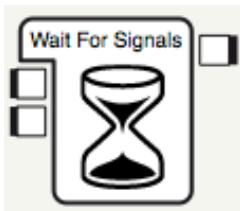
Note : Elle est réinitialisée quand le diagramme contenant la boîte est *unloaded*.



## Boîtier de commutation

Vérifie la valeur de l'*input* et stimule l'*output* correspondant.

S'il n'y a pas d'*output* correspondant, l'*output par défaut* est stimulé.

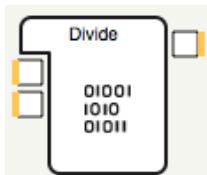


## Attendre les signaux

Attend que les deux *inputs* soient stimulés avant de stimuler l'*output*.

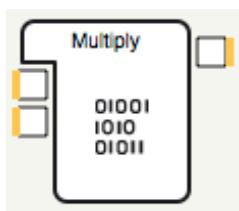
## Math

Ce dossier contient 4 boîtes qui permettent de programmer des logarithmes.



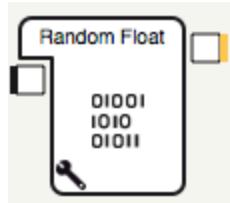
## Diviser

Divise la valeur du premier *input* par la valeur du deuxième *input* et envoie le résultat à l'*output*.



## Multiplier

Multiplie la valeur du premier *input* par la valeur du deuxième *input* et envoie le résultat à l'*output*.

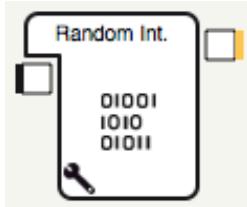


## Nombre en virgule flottante aléatoire

Génère un nombre en virgule flottante (voir ex. ci-dessous) à partir d'une valeur minimale et d'une valeur maximale programmées dans les paramètres).

Ex. :

$$1.3254 = \underbrace{13254}_{\text{mantisse}} \times 10^{-4}^{\text{exposant}}$$

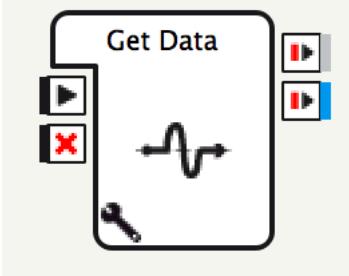


## Nombre entier aléatoire

Génère un nombre entier compris entre un nombre minimal et un nombre maximal, programmés dans les paramètres.

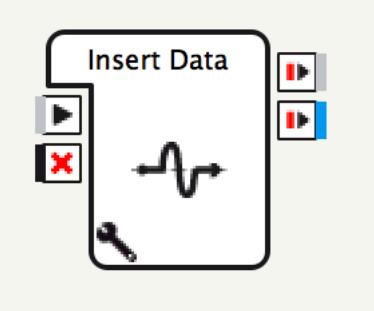
## Memory

Ce dossier contient 6 boîtes qui permettent de gérer la mémoire du robot.



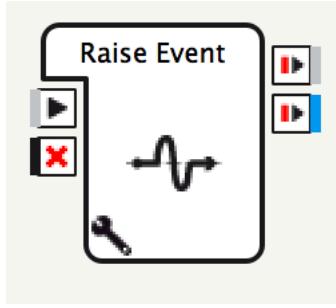
## Récupérer une donnée

Récupère une donnée dans la mémoire partagée du robot.



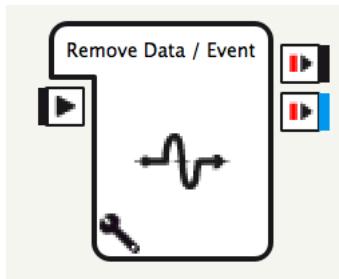
## Stocker une donnée

Stocke dans la mémoire partagée du robot la valeur donnée.



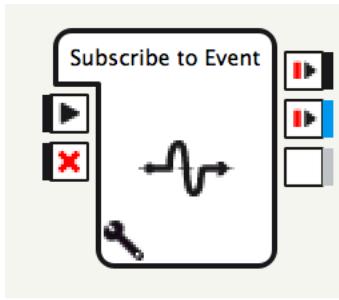
## Diffuser l'évènement

Stocke dans la mémoire partagée du robot la valeur donnée et diffuse l'évènement à tous ses abonnés.



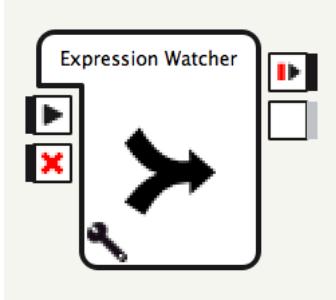
## Supprimer une donnée

Retire une valeur donnée de la mémoire partagée du robot



## S'abonner à un signal

S'abonne à un signal ou un évènement de la mémoire partagée du robot.



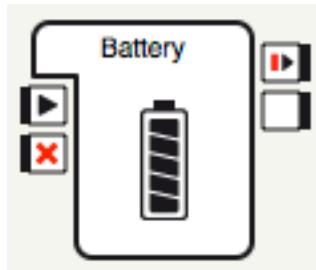
## Observateur d'expression

Cette boîte est une interface pour le module ALEExpressionWatcher. La sortie onExpressionTriggered() sera déclenchée chaque fois que la condition d'expression est valide.



### Robot Information

Ce dossier contient 8 boîtes qui permettent de donner des informations sur le robot



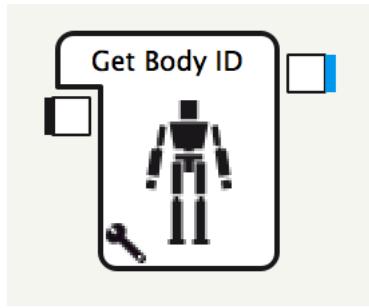
### Piles

Vérifie le détecteur des piles. Va stimuler l'*output onLow* lorsque le niveau des piles est très bas.



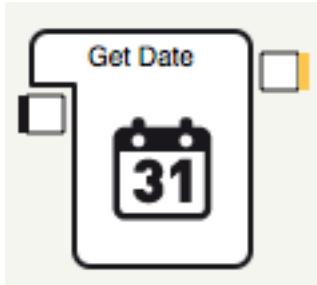
### Vérifier la connexion à *Choregraphe*

Vérifier si le robot est connecté à *Choregraphe* et avertit lors d'une connexion ou déconnexion.



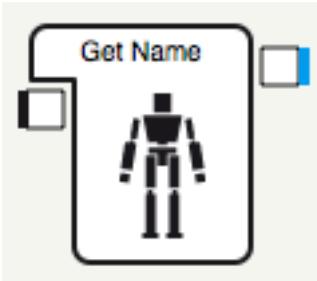
### Obtenir l'identifiant du corps du robot

Renvoie l'identifiant du corps du robot.



## Obtenir la date

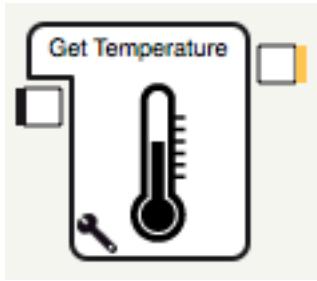
Indique la date et l'heure locale.



## Obtenir le nom

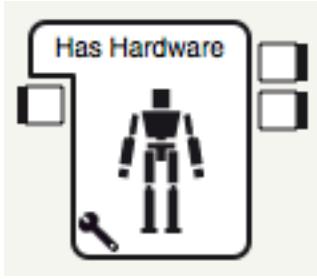
Renvoie le nom du robot.

Note : vous pouvez changer le nom du robot sur sa page Web (adresse IP).



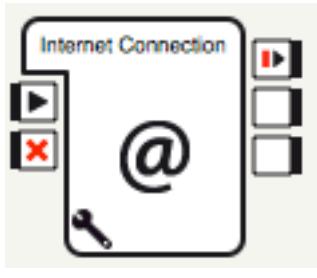
## Obtenir la température

Déetecte la température du matériel (équipement du robot) défini dans les paramètres.



## As-tu ce matériel ?

Si le robot détient le matériel (équipement) défini dans les paramètres, l'*output true* (vrai) est stimulé. Sinon, c'est l'*output false* (faux) qui est stimulé.

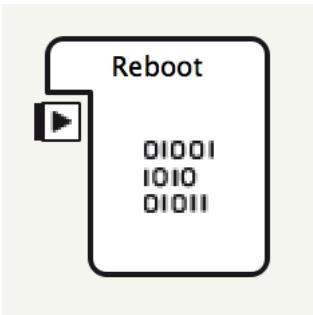


## Vérifier la connexion à Internet

Déetecte une connexion (ou l'absence d'une connexion) à Internet.

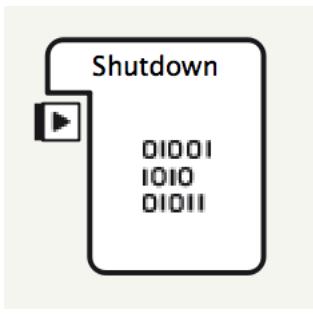
## System Actions

Ce dossier contient 2 boîtes qui permettent de redémarrer ou d'arrêter le système.



### Redémarrer le robot

Redémarre le robot connecté s'il n'est pas virtuel.

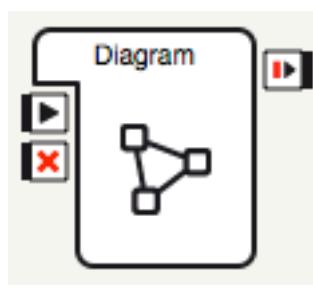


### Arrêter le robot

Arrête le robot s'il n'est pas virtuel.

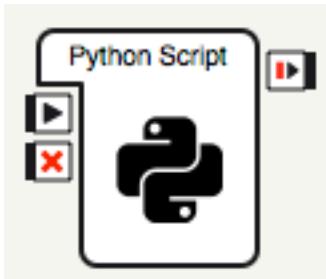
## Templates

Ce dossier contient 2 boîtes qui permettent de créer de nouveaux comportements.



### Diagramme

Cette boîte est vide et devrait être utilisée pour créer une boîte diagramme de votre choix.  
Double-cliquer sur la boîte pour la définir.



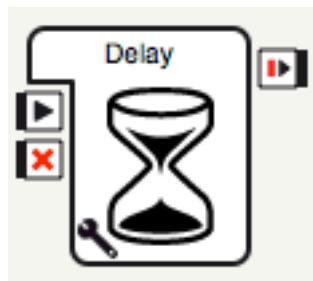
## Langage script *Python*

Cette boîte est vide et devrait être utilisée pour créer une boîte en langage script (*Python*) de votre choix.

Double-cliquer sur la boîte pour la définir.

### Time

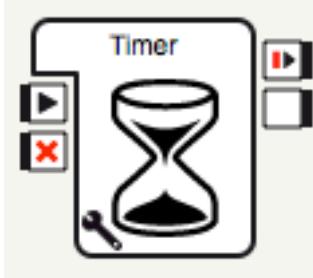
Ce dossier contient 3 boîtes qui permettent de gérer le temps.



## Délai

Attendre un moment avant d'activer l'*output*.

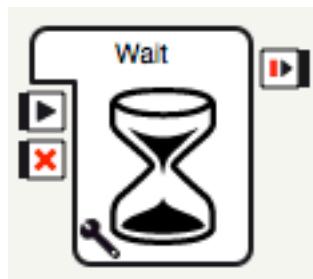
Peut être arrêté à tout moment.  
Plusieurs *inputs* vont déclencher plusieurs *outputs*.



## Minuteur

Stimule la sortie de façon répétitive avec un intervalle spécifié.

Peut être arrêté à tout moment.  
Stimule l'entrée à nouveau et remet le minuteur à zéro.



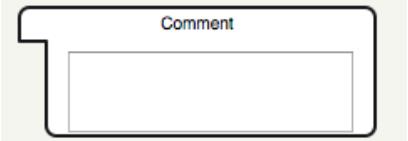
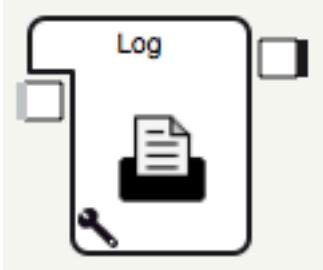
## Attente

Attendre un moment avant d'envoyer le signal à l'*output*.

Peut être arrêté à tout moment.  
Stimuler l'entrée à nouveau avant que la sortie soit activé recommence la période d'attente.

## Tools

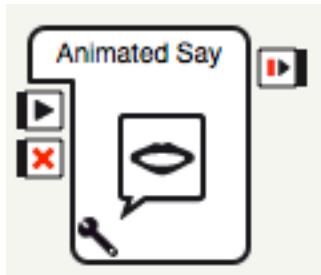
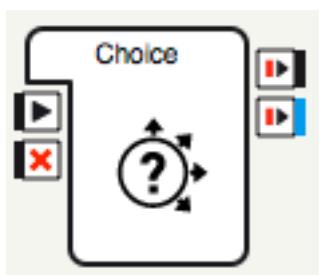
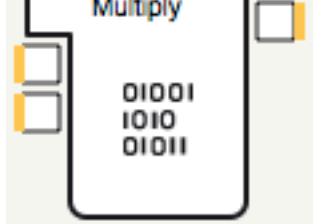
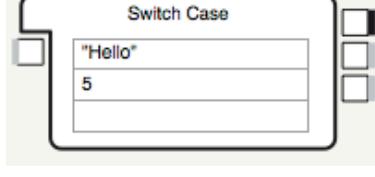
Ce dossier contient 2 outils de Choregraphe.

	<b>Commentaire</b>	Permet de commenter le comportement. Cette boîte n'est pas fonctionnelle et n'a aucun effet sur le comportement.
	<b>Message</b>	Affiche un message dans la fenêtre de débogage.

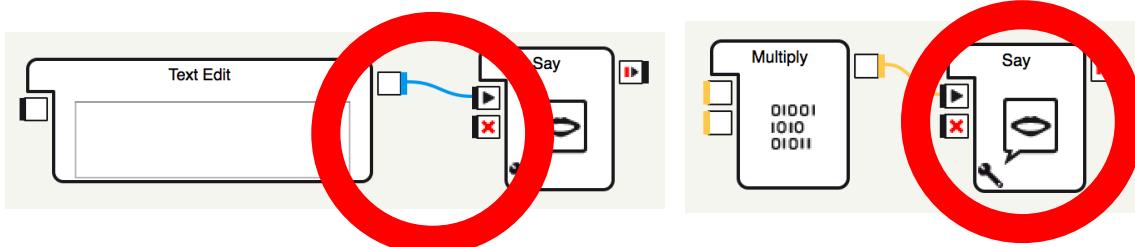
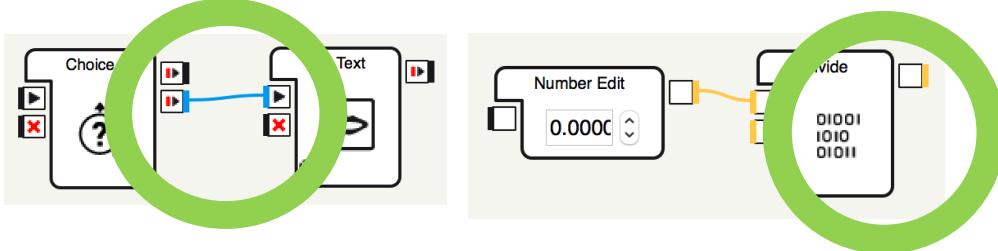
## 2.7. Généralités sur les entrées et les sorties

Les boîtes possèdent des entrées (*input*) et des sorties (*output*). À un niveau plus avancé, le nombre et le type des *inputs* et *outputs* changent.

Il y a **4 types de signaux** qui peuvent être envoyés :

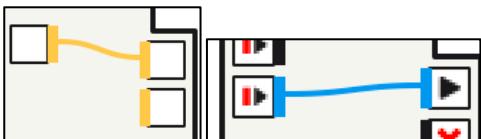
	<b>BANG</b> (noir)	<p>Le signal qui entre et qui sort est seulement un signal, comme un bouton On/Off.</p> <p>Quand le signal entre, il déclenche la boîte. Une fois la boîte terminée, il sort.</p>
	<b>TEXTE</b> (bleu)	<p>Le signal qui entre et/ou qui sort est du texte.</p> <p>Le signal qui va sortir de l'output bleu sera du texte. Tu dois donc utiliser une boîte avec un input bleu pour que cela fonctionne.</p>
	<b>NUMÉRIQUE</b> (jaune)	<p>Le signal qui entre et/ou qui sort est un nombre.</p>
	<b>NEUTRE</b> (gris)	<p>Dans ce type de boîte, tu peux faire entrer et sortir n'importe quel type de signal (BANG, TEXTE ou NUMÉRIQUE).</p>

Voici des exemples de bonnes et de mauvaises connexions :

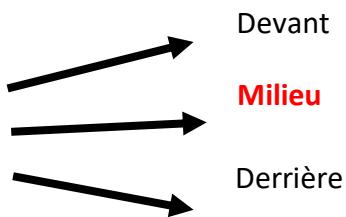
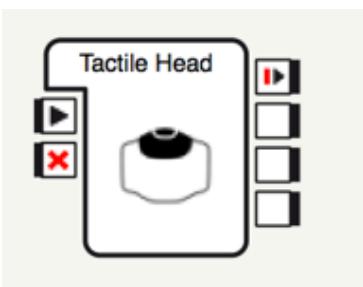


## IMPORTANT

L'entrée, la sortie et le lien (ligne) entre les deux doivent être de la **même couleur** !



Vous devez aussi vous souvenir que les sorties (*outputs*) ont une fonction particulière. Quand il y en a plusieurs, chacune a un rôle différent :



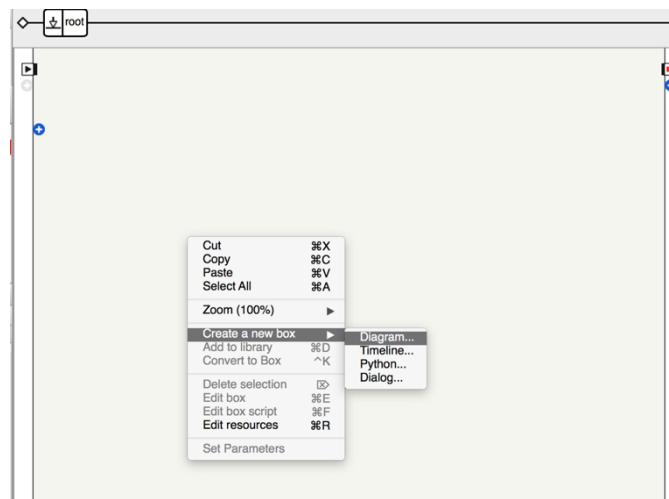
Quand vous allez toucher la tête de NAO au milieu, le signal va sortir de l'***output middle***.

## 2.8. Création de nouvelles boîtes

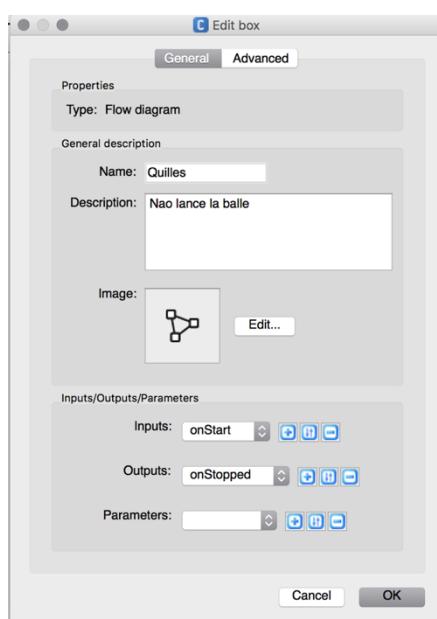
### 2.8.1 Diagram

Une boîte *Diagram* regroupe plusieurs boîtes liées entre-elles et permet de créer un comportement isolé.

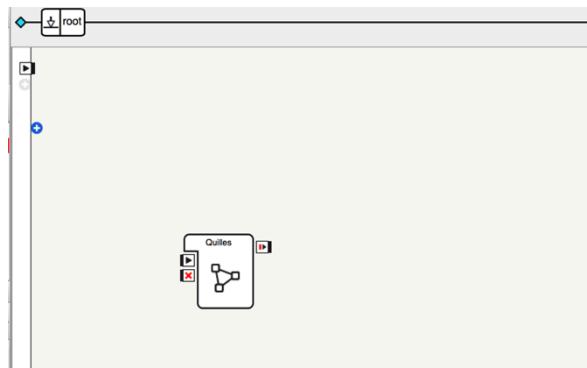
1. Cliquez droit dans la zone de travail et appuyez sur *Create a new box*, puis sur *Diagram*.



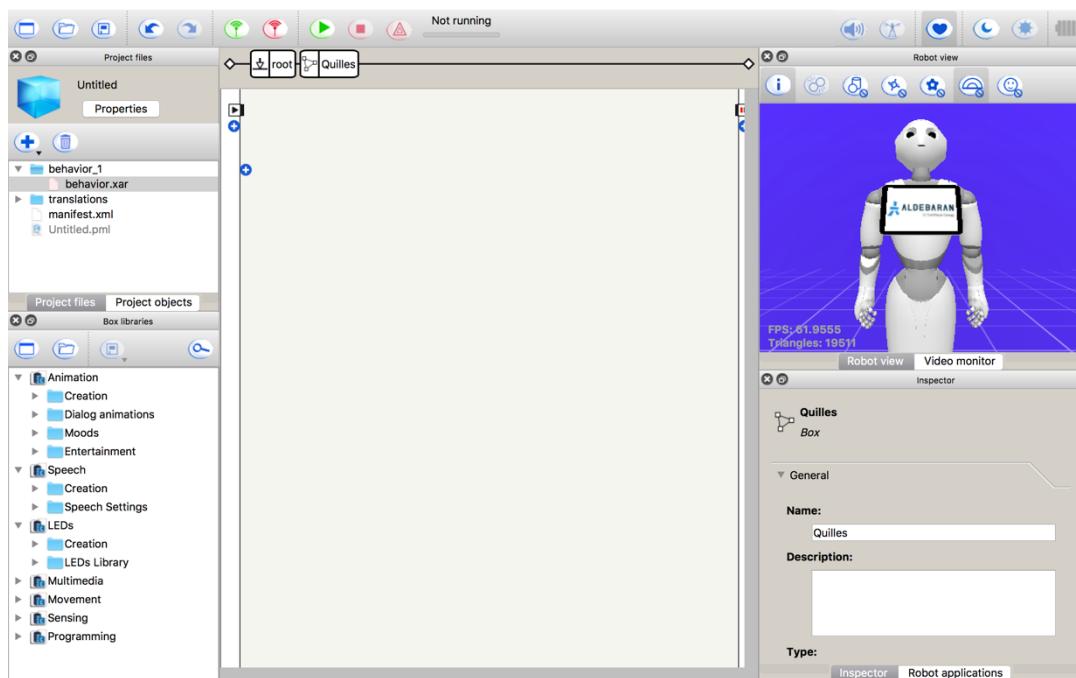
2. Donnez un nom et une description à votre boîte. Vous pouvez aussi définir les paramètres, les entrées et les sorties.



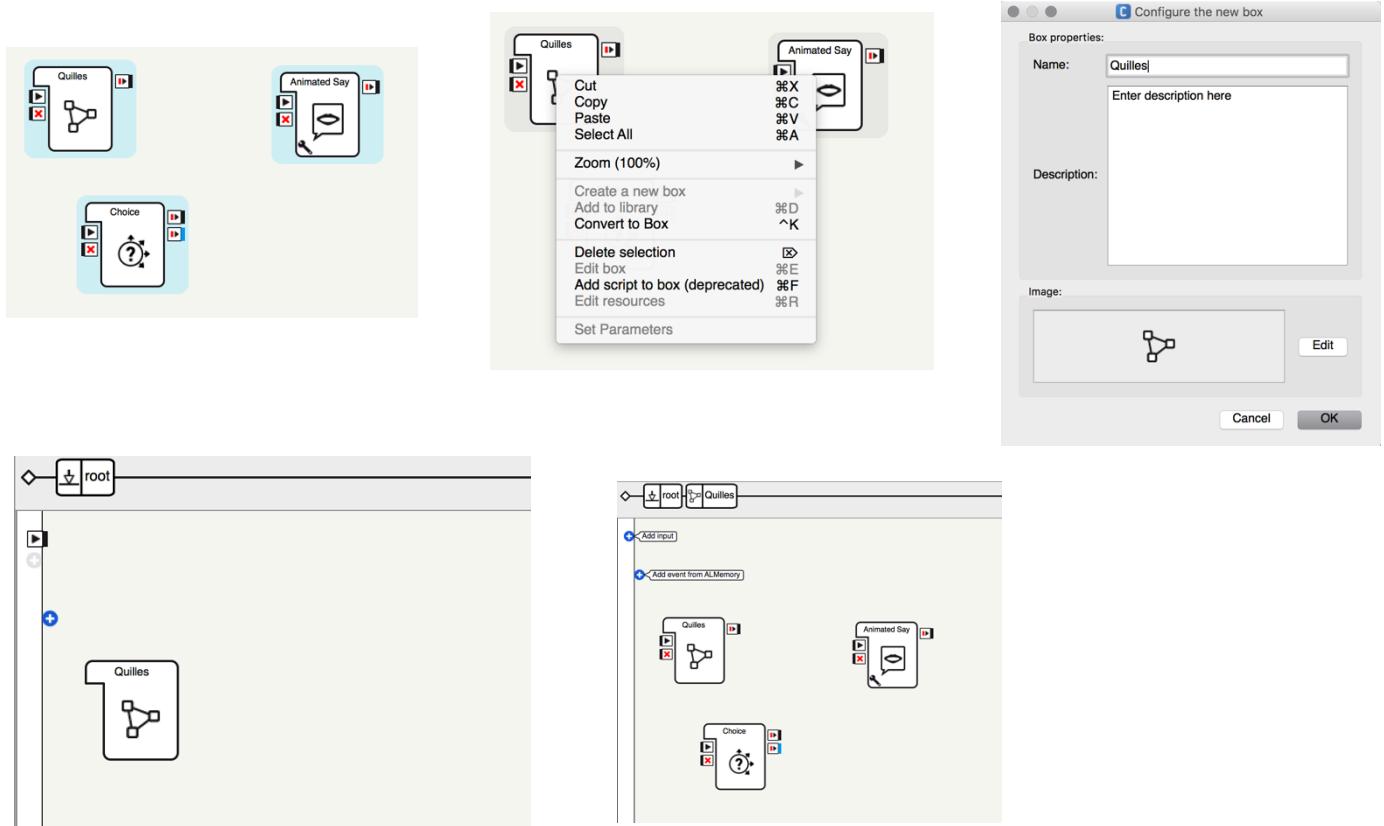
3. Cliquez sur *Ok*. La boîte apparaît dans votre espace de travail.



4. Double-cliquez dessus pour y entrer. Vous pouvez maintenant y ajouter les comportements que vous voulez.



Note : Vous pouvez regrouper plusieurs boîtes entre-elles. Sélectionnez les boîtes, cliquez droit et sélectionnez *Convert box*.

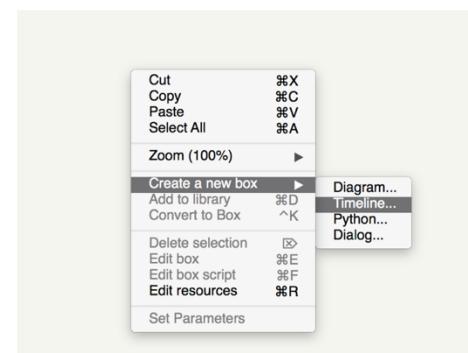


## 2.8.2 Timeline

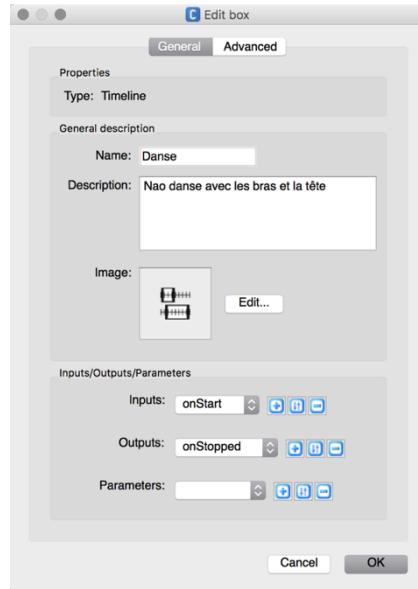
### 2.8.2.1 Créer une boîte Timeline

La boîte *Timeline* est une zone de scénario qui vous permet d'enregistrer des positions sur le robot dans un ordre chronologique. Vous pouvez ainsi créer de nouveaux mouvements.

1. Cliquez droit dans la zone de travail, appuyez sur *Create a new box*, puis sur *Timeline*.



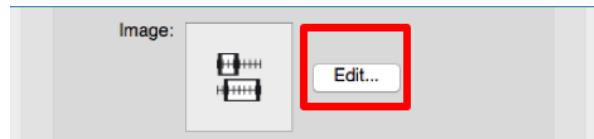
2. Donnez un nom et une description à votre boîte.



3. Si vous voulez, vous pouvez aussi ajouter une image afin que ce soit plus facile de reconnaître vos boîtes dans l'espace de travail.

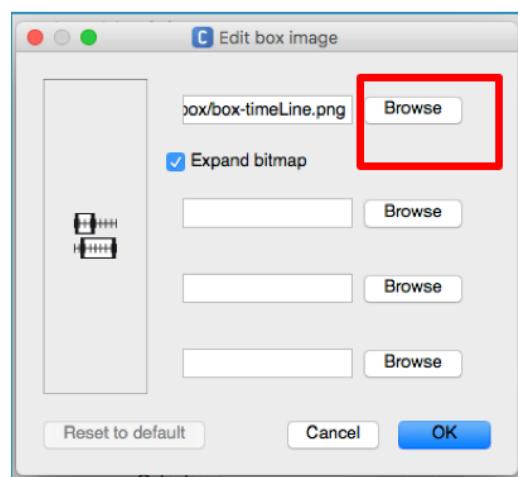
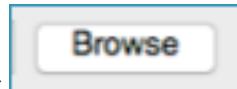
NOTE : Vous pouvez créer une image vous-même au format .png (sur Paint, par exemple), ou la prendre sur Internet.

Appuyez sur :

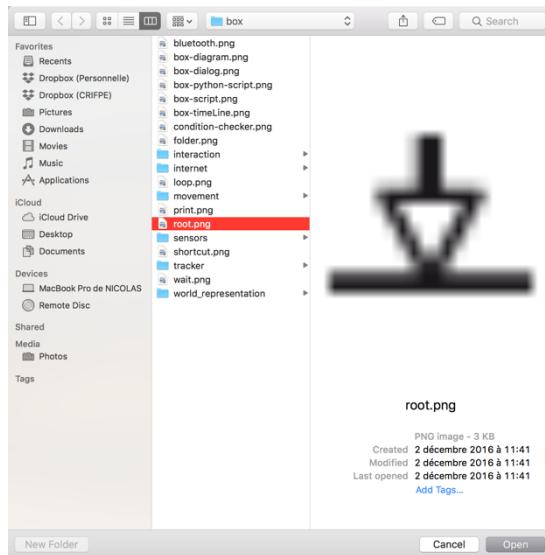


La fenêtre *Edit box image* va apparaître :

Appuyez sur

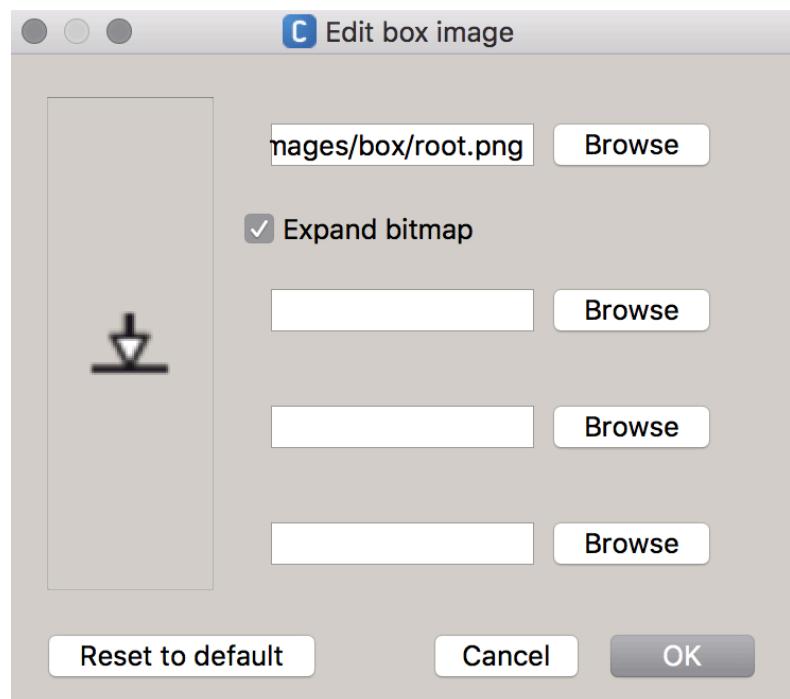


Cherchez dans vos documents l'image que vous avez choisie :



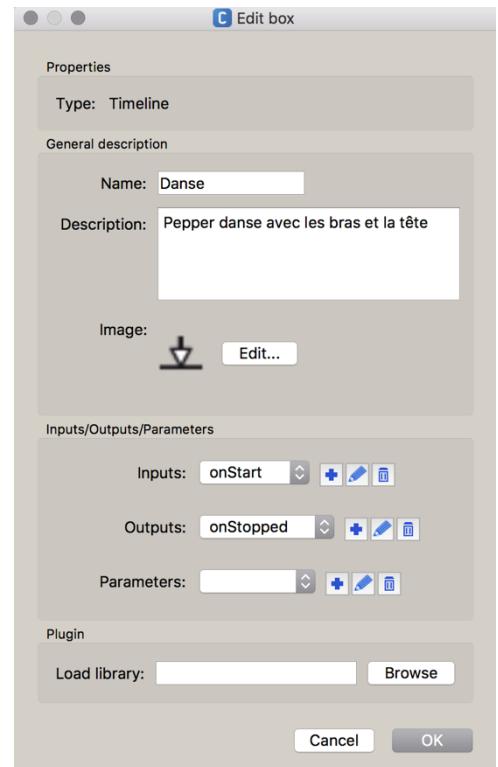
Appuyez sur *Open*.

Votre image apparaît à gauche, dans la fenêtre.



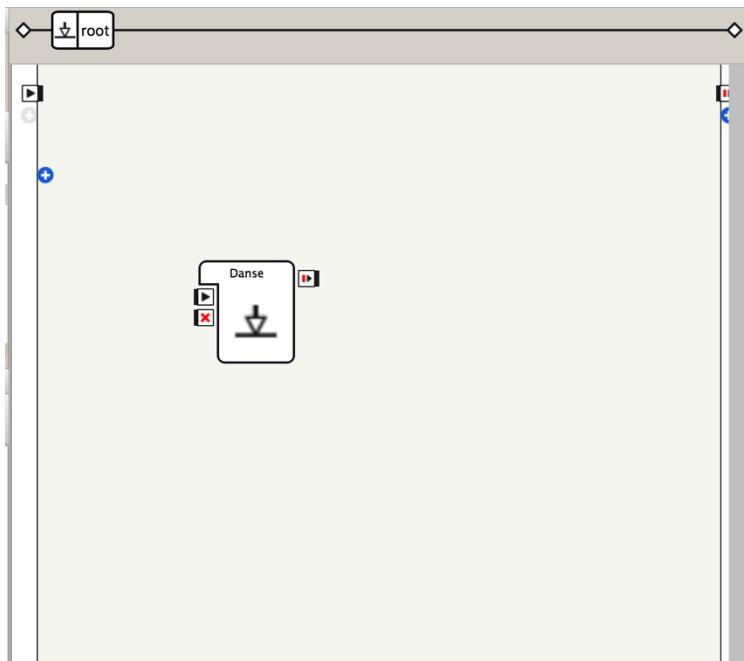
Appuyez sur *OK*.

Vous revenez à la fenêtre *Edit box*.

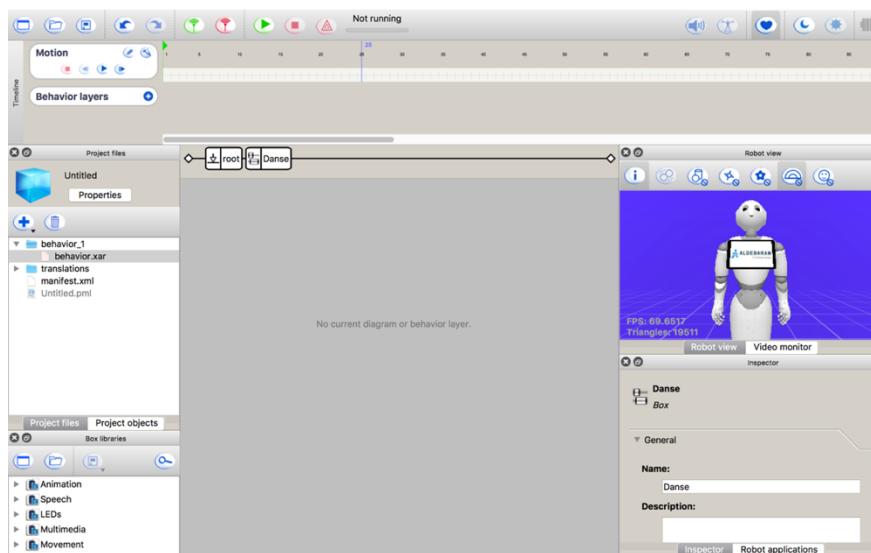


4. Vous pouvez aussi définir les paramètres, les entrées et les sorties.

5. Cliquez sur *Ok*. La boîte apparaît dans votre espace de travail.



6. Double-cliquez dessus pour y entrer. Vous pouvez maintenant éditer le scénario.



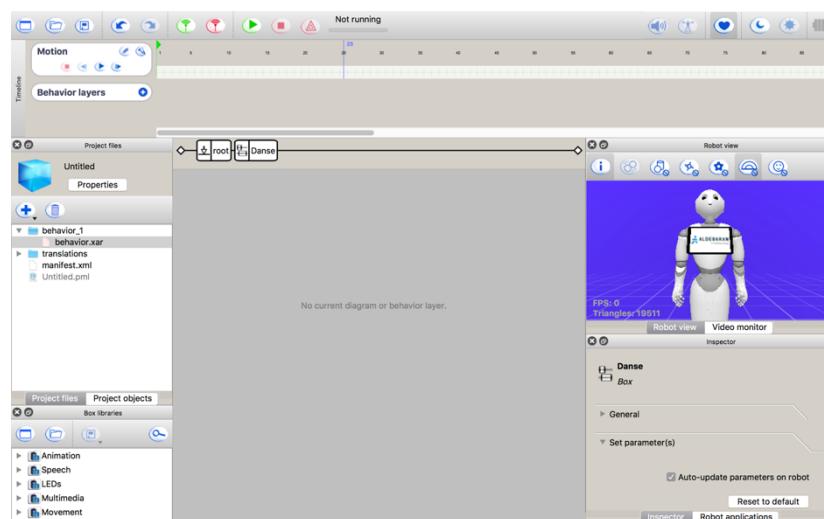
### 2.8.2.2 Création d'un scénario avec le mode Animation

Maintenant que vous savez créer et personnaliser une boîte *Timeline*, vous allez apprendre à créer un mouvement.

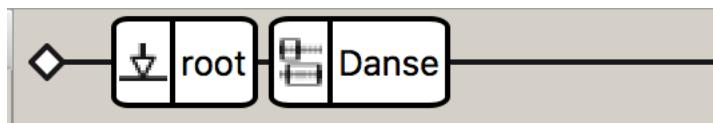
1. Pour cela, vous devez double-cliquer sur la boîte.

Pour cet exemple, nous allons continuer avec la boîte *Danse*.

Double-cliquez sur la boîte et vous arrivez sur cette page :



Vous pouvez voir que vous êtes dans la boîte *Danse* en regardant ici :



Vous allez maintenant créer les différentes positions de votre mouvement « Danse » en bougeant directement les membres du vrai robot qui sera en mode *animation*.



2. D'abord, désactivez la vie autonome du robot en cliquant sur  ou en cliquant deux fois sur le bouton *Power*.



3. Réveillez votre robot en appuyant sur le bouton . Si vous ne pouvez pas appuyer sur le bouton, c'est que votre robot est déjà réveillé.



4. Appuez sur le bouton d'animation . Le bouton va devenir rouge, ce qui veut dire que le mode *animation* est activé.

5. Vous pouvez maintenant manipuler le vrai robot en utilisant les capteurs tactiles du robot pour relâcher la tension.

Lorsque vous passez en mode animation, les yeux de PEPPER deviennent rouges. Les moteurs sont sous tension. Lorsque vous souhaitez relâcher la tension d'un des membres pour créer une position, vous devez appuyer sur l'un des capteurs tactiles. Le voyant correspondant deviendra alors vert.

Pour manipuler la tête, vous devez appuyer une fois sur le capteur central de la tête. Le haut des yeux est alors vert. Pour le remettre sous tension, vous devez appuyer à nouveau sur le capteur central.

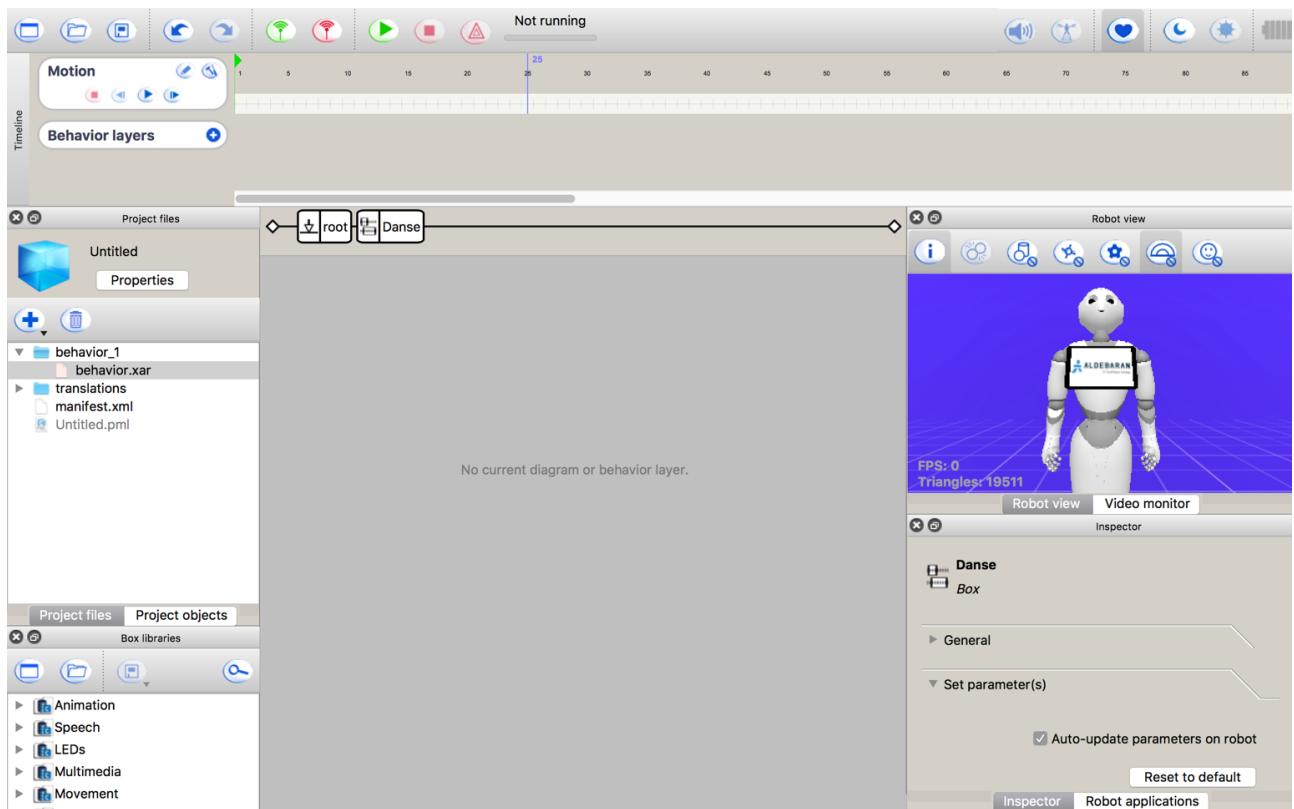
Pour manipuler les bras, vous devez rester appuyé sur le capteur sensoriel situé sur la main du bras que vous souhaitez bouger. Lorsque vous lâchez le capteur, le robot se remet sous tension.

Vous avez deux solutions pour enregistrer les positions : soit vous placez des marques sur la ligne de temps de votre boîte *Timeline*, soit que vous appuyez simultanément sur les trois capteurs tactiles de la tête

### 2.8.2.3 Crédit d'un marqueur sur la ligne de temps

En mode *Animation*, vous allez manipuler le vrai robot et vous allez apprendre à enregistrer des positions en créant des marques sur la ligne de temps. Pour cet exemple, nous continuons avec la boîte *Timeline Danse*.

Dans le haut de la fenêtre, vous apercevez la ligne du temps (la *Timeline*).



Vous devez créer des marques pour chaque position.

Pour enregistrer la première position (position de départ), vous allez créer une première marque à 10.

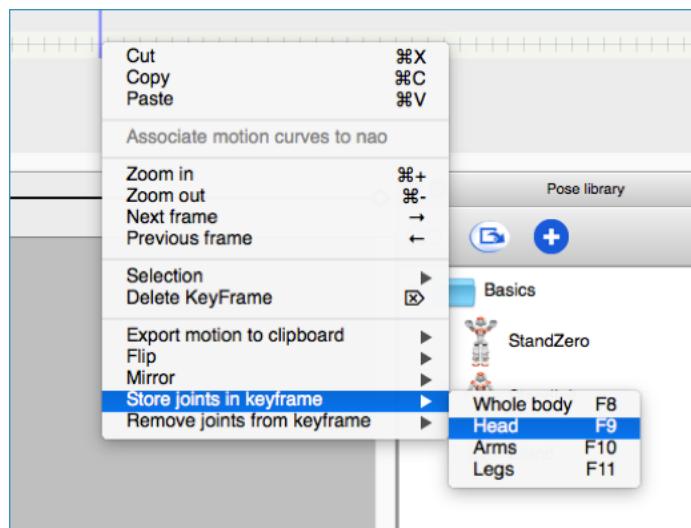
1. Placez votre curseur sur la ligne de temps au niveau du 10 :



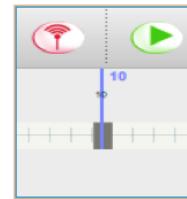
2. Cliquez droit et sélectionnez *Store joints in keyframe*.

4 options apparaissent : *Whole body*, *Head*, *Arms* et *Legs*. Si vous décidez de créer une marque pour le mouvement d'un seul membre, sélectionnez l'option correspondante. Si vous voulez enregistrer le mouvement de plusieurs membres, sélectionnez *Whole body*.

3. Dans cet exemple, sélectionnez *Head*.



4. Vous allez voir apparaître un petit rectangle gris foncé sur ta ligne du temps : c'est votre marque. Assurez-vous que votre repère (ligne bleue) soit bien situé sur 10.



Pour la deuxième position :

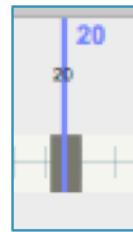
5. Tenez bien votre robot par la taille/le torse pour éviter qu'il ne tombe.

6. Appuize sur le capteur tactile au milieu de la tête pour relâcher la tension de la tête.

7. Tournez la tête de NAO vers la gauche.

8. Appuez sur le capteur tactile au milieu de la tête pour rétablir la tension.

9. Créez une nouvelle marque à 20 en suivant la même procédure. Assurez-vous que votre repère (ligne bleue) soit bien sur la marque à 20.



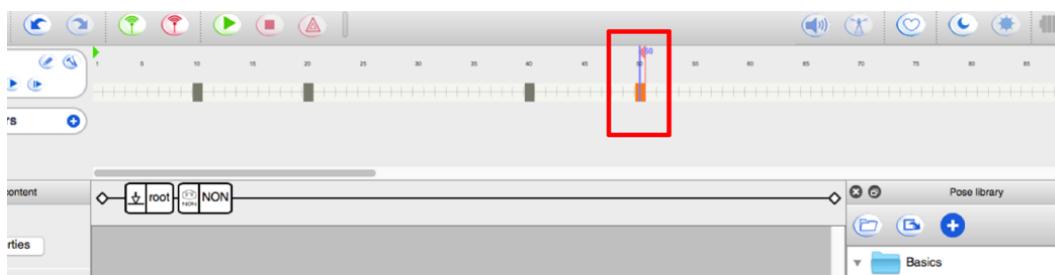
Donc à 10, la tête est au centre et à 20, elle est à gauche.

Vous allez maintenant faire la même opération pour bouger les bras.

9. À 40, vous levez le bras droit du robot.



10. À 50, vous redescendez le bras droit du robot.



Vous devriez donc avoir 4 marques sur votre ligne de temps :

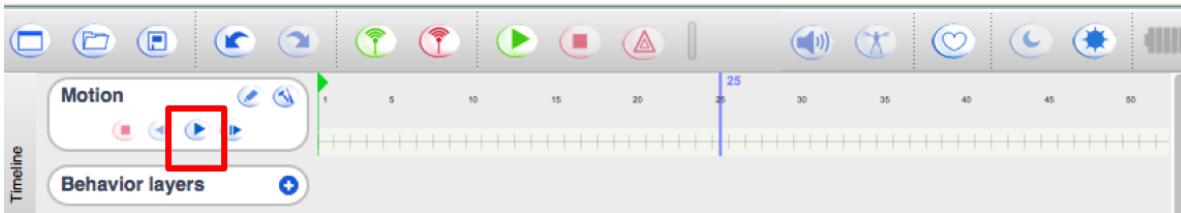


11. La création du mouvement est terminée. Vous pouvez maintenant désactiver le mode animation

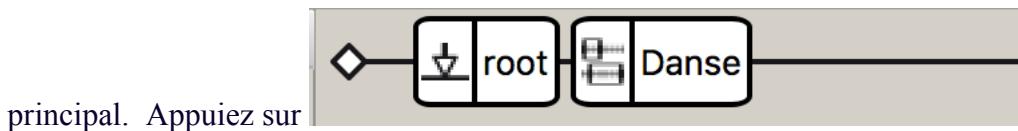
en cliquant sur le bouton suivant :



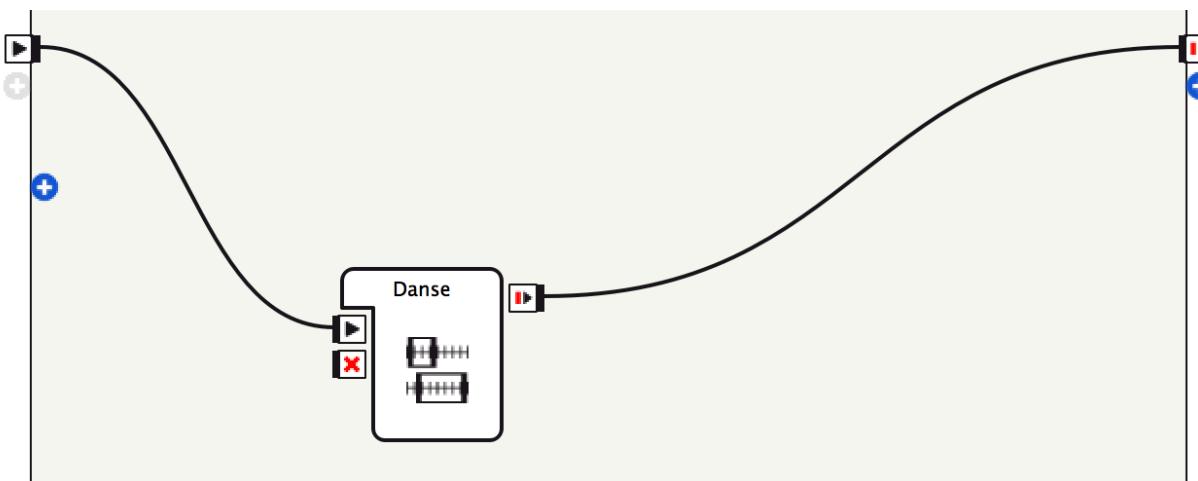
12. Pour tester le comportement, appuyez sur le bouton lecture :



13. Une fois que vous avez créé un mouvement, vous pouvez revenir à votre espace de travail



principal. Appuez sur



15. Appuyez sur pour tester ta séquence.

**Note** : Grâce à cette première méthode d'enregistrement de positions, vous pouvez également créer un mouvement via la fenêtre *Robot View*. Cela peut vous servir si vous n'avez pas de vrai robot à disposition et que vous souhaitez vous entraîner avec le robot virtuel. Mais attention, la manière d'enregistrer sur la ligne de temps diffère légèrement. De plus, vous n'avez pas besoin de cliquer sur le bouton *Animation* ou sur le bouton *Réveil*.

- D'abord enregistrez la position de base de la même façon qu'avec un robot réel. Vous placez votre curseur à l'endroit souhaité sur la ligne de temps.
- Vous cliquez-droit, vous sélectionnez *Store joints in keyframe* et vous choisissez l'option désirée. Un petit rectangle gris apparaît sur la ligne de temps.
- Maintenant, cliquez plus loin sur la ligne de temps. Un trait bleu apparaît.
- Allez sur la fenêtre *Robot View*, et placez votre robot dans la position désirée. Un petit rectangle gris apparaîtra automatiquement sur la ligne de temps, à l'endroit où se situait le trait bleu.

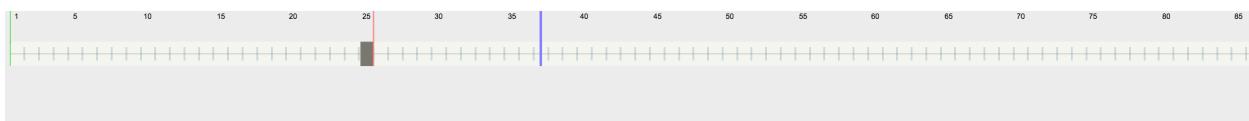
Du fait de l'enregistrement automatique des positions sur la ligne de temps, pensez bien à placer plus loin votre curseur, avant de donner une nouvelle posture à votre robot.

De plus, il est très difficile d'être précis via cette fenêtre, donc si vous souhaitez tester plus tard votre mouvement, sur un vrai robot, il risque de tomber. Fais donc des gestes très simples et espaces les marques de temps.

#### 2.8.2.4 Enregistrement des positions sur le robot

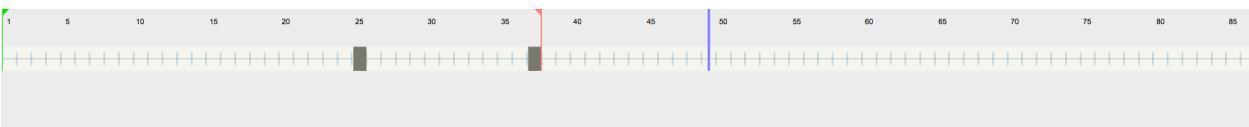
ATTENTION : Pensez à vérifier que la vie autonome du robot est bien désactivée pour utiliser cette méthode. Autrement, votre robot sortira du mode *Animation* lorsque vous enregistrerez une position en appuyant sur sa tête.

1. Pour enregistrer la première position (position de départ), appuez brièvement sur l'ensemble des capteurs tactiles de la tête du robot, avec votre main. Une marque va apparaître sur la ligne de temps et le marqueur bleu se place automatiquement plus loin. À chaque fois que NAO enregistre une position, ses oreilles émettent une lumière bleue.



Pour la deuxième position :

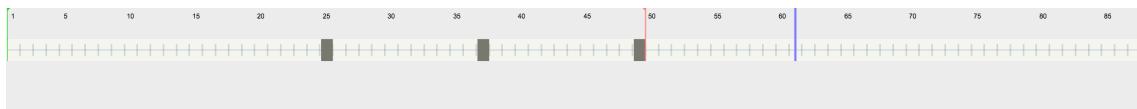
2. Tenez bien votre robot par la taille/le torse pour éviter qu'il ne tombe.
3. Appuez sur le capteur tactile central de sa tête pour relâcher la tension.
4. Tournez la tête vers la gauche.
5. Appuez brièvement sur l'ensemble des capteurs tactiles de la tête pour enregistrer la position. Une deuxième marque va apparaître sur la ligne de temps. Attention ! En appuyant sur la tête, vous avez rétabli la tension.



Donc, pour la troisième position :

6. Appuez sur le capteur tactile central de sa tête pour relâcher à nouveau la tension.
7. Levez le bras droit.

8. Appuez brièvement sur l'ensemble des capteurs tactiles de la tête pour enregistrer la position. Une nouvelle marque va apparaître sur la ligne de temps.



9. Pour la quatrième position et dernière position redescendez le bras droit.



11. La création du mouvement est terminée. Vous pouvez maintenant désactiver le mode *animation*

en cliquant sur le bouton suivant :

12. Testez le comportement pour vérifier que NAO fait bien « NON » de la tête.

## 2.8.3 Python

### 2.8.3.1 Qu'est-ce que Python ?

Le langage de programmation dans Choregraphe s'appelle Python. Chaque boîte est un module. En résumé, un module est comme une application, complètement indépendante, qui peut fonctionner seule ou qui communique facilement avec les autres modules.

Les codes ressemblent à ceci :

```
class MyClass(GeneratedClass):
    def __init__(self):
        GeneratedClass.__init__(self)

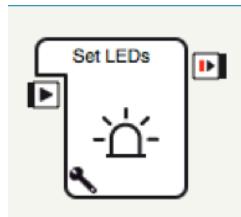
    def onInput_onStart(self):
        self.onStopped([252, 0, 0])
```

Toutes les boîtes dans Choregraphe ont leur propre codage, alors tu peux facilement les éditer. Vous remarquerez qu'elles commencent toutes par une première ligne « Class MyClass (GeneratedClass) ».

*MyClass* est une classe de modules générée automatiquement dans Choregraphe. Elle contient toutes les informations nécessaires des boîtes telles que les entrées, les sorties et les paramètres ainsi que d'autres fonctions complémentaires qui pourraient être intégrées dans le script (chronologie, charger ou décharger un événement, ressources, proxies, etc.).

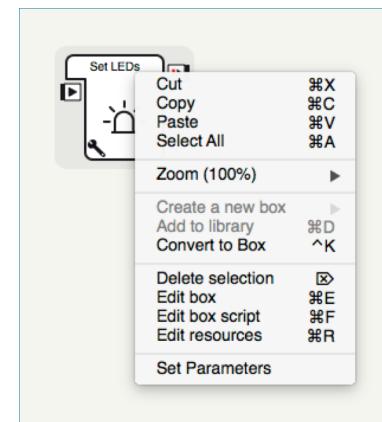
Il faut être vigilant lorsque l'on modifie le codage d'une boîte, puisqu'un seul symbole manquant ou à la mauvaise place peut entraîner un dysfonctionnement du programme. Cela demande un réel apprentissage et de l'entraînement. À un niveau plus avancé, il est également possible de programmer des modules Python à l'extérieur du logiciel Choregraphe et de les importer.

#### 2.8.3.2 Description de la programmation d'une boîte



Prenons en exemple la boîte *Set LEDs*.

Pour voir le programme de la boîte, cliquez-droit dessus et appuyez sur *Edit box script* (ou CMD+F) :



```

1 class MyClass(GeneratedClass):
2     def __init__(self):
3         GeneratedClass.__init__(self, False)
4
5     def onLoad(self):
6         self.leds = ALProxy("ALLeds")
7
8     def onUnload(self):
9         #~ puts code for box cleanup here
10        pass
11
12     def onInput_onSet(self):
13         self.leds.fade(self.getParameter("LEDs group"),
14                     self.getParameter("Intensity (%))/100.,
15                     self.getParameter("Duration (s)"))
16         self.onReady() # activate output of the box

```

The screenshot shows a Mac OS X-style window titled "Script editor" with a tab labeled "Set LEDs". Inside the editor, there is a code block written in Python. The code defines a class named "MyClass" that inherits from "GeneratedClass". It contains four methods: "\_\_init\_\_", "onLoad", "onUnload", and "onInput\_onSet". The "onInput\_onSet" method uses the "ALProxy" object "self.leds" to fade the LEDs based on parameters from the configuration file.

La fenêtre *Script editor* va apparaître :

Quand vous écrivez le programme, vous devez vous souvenir que vous êtes en train de définir le fonctionnement de la classe « *MyClass* ».

Dans le programme ci-dessous, il y a 4 méthodes :

1. *Initialize*

2. *onLoad*

3. *onUnload*

4. *onInput\_onSet*

```

1 class MyClass(GeneratedClass):
2     def __init__(self):
3         GeneratedClass.__init__(self, False)
4
5     def onLoad(self):
6         self.leds = ALProxy("ALLeds")
7
8     def onUnload(self):
9         #~ puts code for box cleanup here
10        pass
11
12     def onInput_onSet(self):
13         self.leds.fade(self.getParameter("LEDs group"),
14                     self.getParameter("Intensity (%))/100.,
15                     self.getParameter("Duration (s)"))
16         self.onReady() # activate output of the box

```

This screenshot is similar to the one above, showing the "Script editor" window with the "Set LEDs" tab. However, the method names "\_\_init\_\_", "onLoad", "onUnload", and "onInput\_onSet" are highlighted with red rectangular boxes, likely to draw attention to them as the focus of the exercise.

Voyons comment les méthodes sont exécutées, les unes après les autres.

```
1 class MyClass(GeneratedClass):
2     def __init__(self):
3         GeneratedClass.__init__(self, False) 1
4
5     def onLoad(self):
6         self.leds = ALProxy("ALLeds") 2
7
8     def onUnload(self):
9         #~ puts code for box cleanup here
10        pass
11
12     def onInput_onSet(self):
13         self.leds.fade(self.getParameter("LEDs group"),
14                     self.getParameter("Intensity (%))/100.,
15                     self.getParameter("Duration (s)"))
16         self.onReady() # activate output of the box 4
```

1. Quand le comportement commence, la méthode **initialize** est exécutée en premier.

```
1 class MyClass(GeneratedClass):
2     def __init__(self):
3         GeneratedClass.__init__(self, False)
4
5     def onLoad(self):
6         self.leds = ALProxy("ALLeds")
```

C'est une fonction qui réinitialise les paramètres de base de la boîte (qui sont communs à toutes les boîtes dans Choregraphe).

2. La méthode **onLoad** : cette méthode est exécutée quand le diagramme de flux de la boîte est en chargement. Elle est nécessairement exécutée après la méthode *initialize*.

```
1 class MyClass(GeneratedClass):
2     def __init__(self):
3         GeneratedClass.__init__(self, False)
4
5     def onLoad(self):
6         self.leds = ALProxy("ALLeds") 1
7
8     def onUnload(self):
9         #~ puts code for box cleanup here
10        pass
```

3. La méthode **onUnload** : cette méthode est exécutée quand le diagramme de flux de la boîte est en déchargement.

```
4
5 def onLoad(self):
6     self.leds = ALProxy("ALLeds")
7
8 def onUnload(self):
9     #~ puts code for box cleanup here
10    pass
11
12 def onInput_onSet(self):
13     self.leds.fade(self.getParameter("LEDs group"),
14     self.getParameter("Intensity (%)")/100.,
15     self.getParameter("Duration (s)"))
```

Après le déchargement, les boîtes sont désactivées et ne peuvent plus recevoir d'événements ou d'informations (inputs).

N.B. : Cette méthode arrête habituellement tout ce qui est actif dans le programme. Voilà pourquoi l'input *onStop* exécute *onUnload* par défaut.



ATTENTION : Ces 3 premières étapes sont obligatoires pour toutes les boîtes !

4. La méthode **onInput\_onSet** : cette méthode est exécutée quand le *onStart* est stimulé. Elle exécute ce qui est en rose dans l'image ci-dessous : *self.leds.fade(self.getParameter(\*LEDs group\*), self.getParameter ('Intensity (%)')/100, self.getParameter ('Duration (s)'))*.

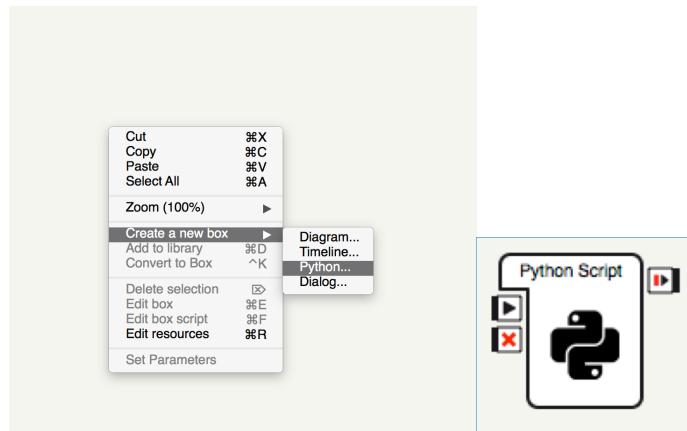
Cet appel au module *self.leds* ordonne au groupe de *LEDs* visé de s'allumer, à un temps donné, à une intensité donnée. À la fin de cette méthode, *self.onReady()* stimule le output *onStopped* de la boîte *Switch LEDs*.

```
7
8 def onUnload(self):
9     #~ puts code for box cleanup here
10    pass
11
12 def onInput_onSet(self):
13     self.leds.fade(self.getParameter("LEDs group"),
14     self.getParameter("Intensity (%)")/100.,
15     self.getParameter("Duration (s)"))
16     self.onReady() # activate output of the box
```

### 2.8.3.3 Créer une nouvelle boîte Python pour faire parler PEPPER

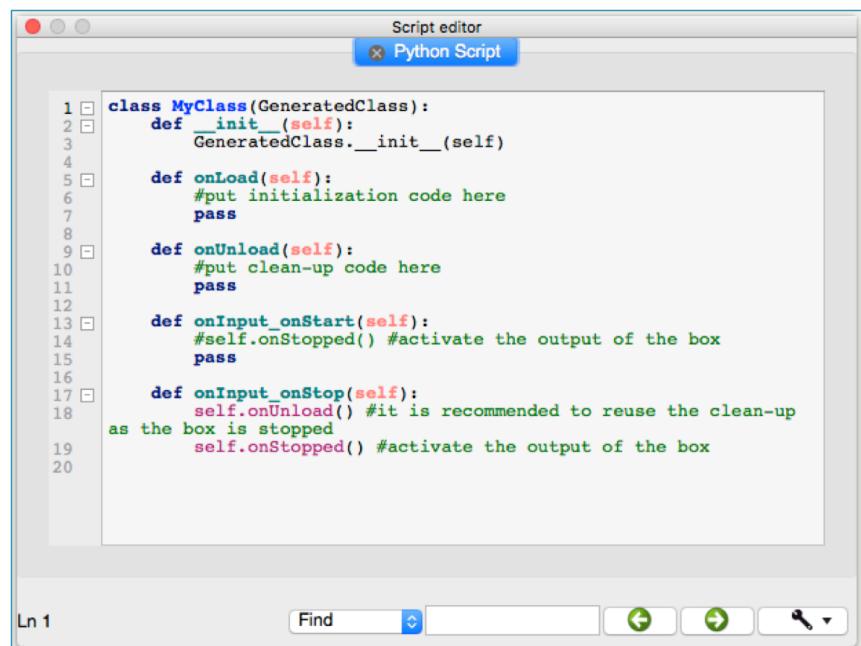
1. Vous devez d'abord ajouter une boîte *Python Script* sur votre espace de travail.

Pour cela, cliquez-droit sur la zone grise et sélectionnez *Create New Box*, puis *Python*. Vous pouvez lui donner un nom, une description et une nouvelle image.



2. Vous pouvez maintenant double-cliquer sur la boîte pour accéder au programme de la boîte.

Cette fenêtre va apparaître :



```
1 class MyClass(GeneratedClass):
2     def __init__(self):
3         GeneratedClass.__init__(self)
4
5     def onLoad(self):
6         #put initialization code here
7         pass
8
9     def onUnload(self):
10        #put clean-up code here
11        pass
12
13     def onInput_onStart(self):
14        #self.onStopped() #activate the output of the box
15        pass
16
17     def onInput_onStop(self):
18        self.onUnload() #it is recommended to reuse the clean-up
19        as the box is stopped
20        self.onStopped() #activate the output of the box
```

3. Vous devez ajouter une ligne de code après la 3e ligne :

```
Script editor
Python Script

1 class MyClass(GeneratedClass):
2     def __init__(self):
3         GeneratedClass.__init__(self)
4
5     def onUnload(self):
6         #put initialization code here
7         pass
8
9
10
11
12
13
14
15
16
17
18
19
20
21
```

```
class MyClass(GeneratedClass):
    def __init__(self):
        GeneratedClass.__init__(self)
        self.tts = ALProxy('ALTextToSpeech')
```

4. Vous devez maintenant ajouter une ligne de code après la 15e ligne :

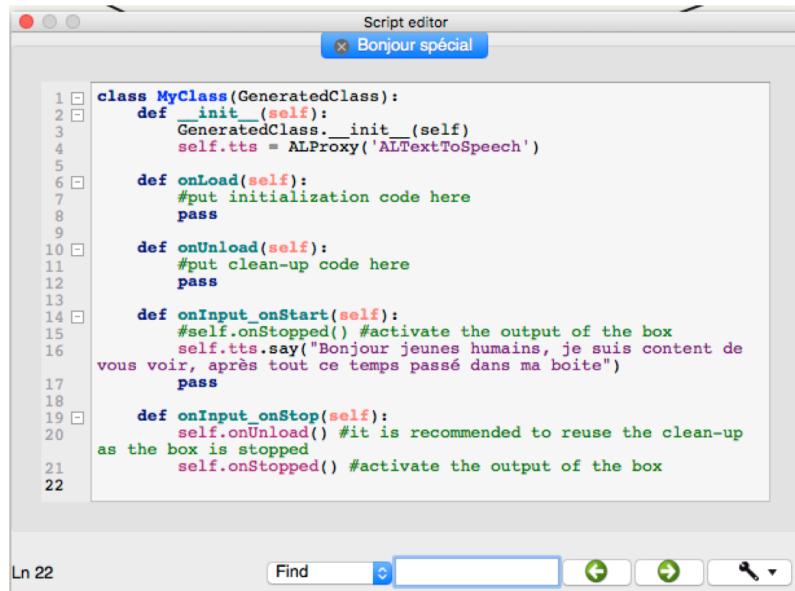
```
13
14     def onInput_onStart(self):
15         #self.onStarted() #activate the output of the box
16         pass
17
18     def onInput_onStop(self):
19         self.onUnLoad() #it is recommended to reuse the clean-up
20             as the box is stopped
21                 self.onStopped() #activate the output of the box
```

Voici le texte à ajouter :

```
def onInput_onStart(self):
    #self.onStarted() #activate the output of the box
    self.tts.say("Bonjour jeunes humains, je suis content de vous voir, après tout ce temps passé dans ma boîte")
    pass
```

Ici, PEPPER va dire : « Bonjour jeunes humains, je suis content de vous voir, après tout ce temps passé dans ma boîte ».

Voilà donc à quoi va ressembler votre fenêtre de codage à la fin :



```
1 class MyClass(GeneratedClass):
2     def __init__(self):
3         GeneratedClass.__init__(self)
4         self.tts = ALProxy('ALTextToSpeech')
5
6     def onLoad(self):
7         #put initialization code here
8         pass
9
10    def onUnload(self):
11        #put clean-up code here
12        pass
13
14    def onInput_onStart(self):
15        #self.onStopped() #activate the output of the box
16        self.tts.say("Bonjour jeunes humains, je suis content de
17        vous voir, après tout ce temps passé dans ma boite")
18        pass
19
20    def onInput_onStop(self):
21        self.onUnload() #it is recommended to reuse the clean-up
22        as the box is stopped
23        self.onStopped() #activate the output of the box
```

Assurez-vous de bien vérifier qu'il ne manque aucune lettre et que les symboles sont les bons, surtout aux endroits suivants :

( 'ALTextToSpeech' )

ET

( "Bonjour jeunes hu  
;sé dans ma boite" )

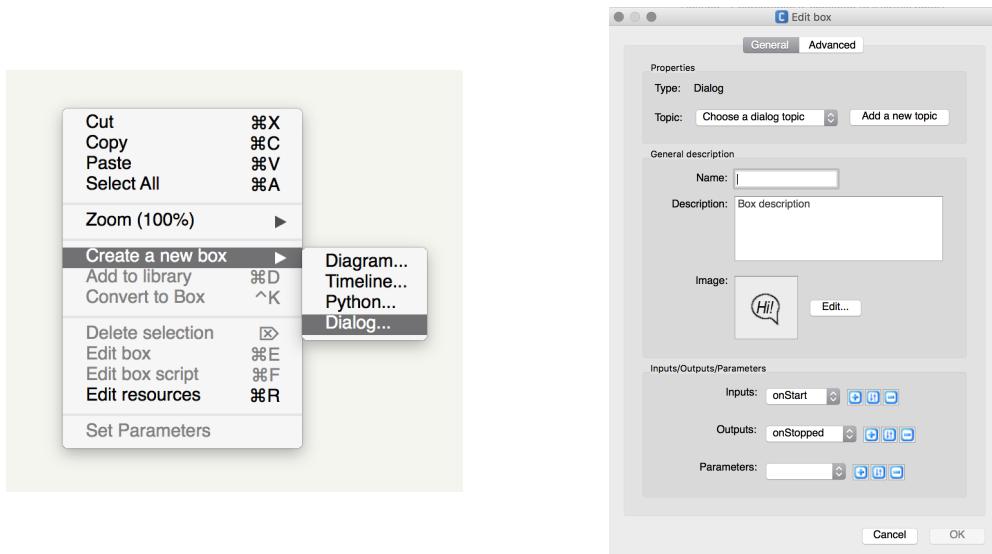
## 2.8.4 Dialog

La boîte *Dialog* permet de créer des conversations. Vous pouvez donner à PEPPER des règles associant des entrées humaines aux sorties du robot. Vous pouvez utiliser un vrai robot ou le robot virtuel.

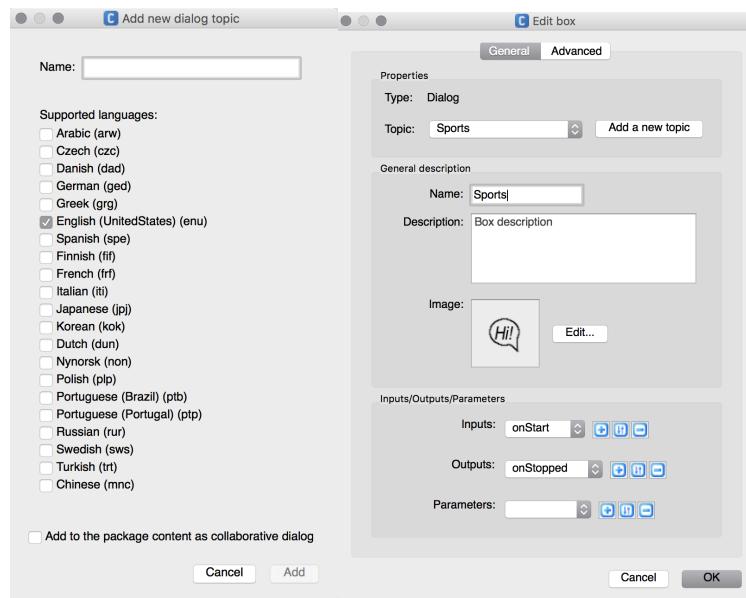
1. Commencez par créer une boîte *Dialog*.

Cliquez-droit dans votre espace de travail. Sélectionnez *Create a new box*, puis, *Dialog*. La fenêtre *Edit Box* apparaît.

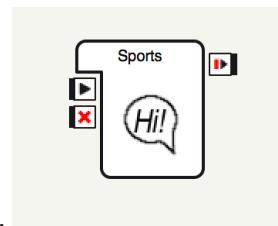
2. Cliquez sur *Add a new topic* ou choisissez un sujet déjà existant dans le menu déroulant.



Une nouvelle fenêtre apparaît. Donnez un nom au sujet, choisissez une langue, cliquez sur *Add*, puis sur *OK*.



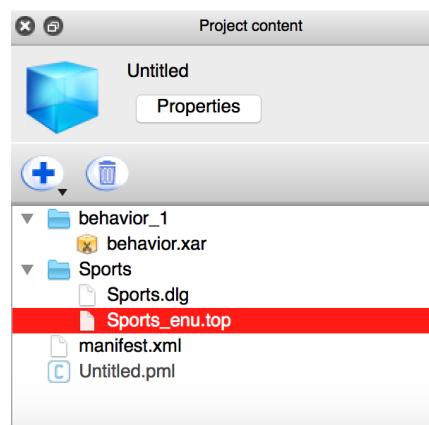
La boîte va apparaître dans la zone de travail. Il ne faut pas d'espace dans le nom du sujet, sinon la boîte ne fonctionnera pas.



Le nom de la boîte est automatiquement complété en créant un nouveau sujet.

Maintenant, vous allez devoir écrire les règles dans la fenêtre *Script Editor*. Le sujet que vous veznez de créer apparaît sous forme d'un dossier dans la fenêtre *Project content*.

3. Double-cliquez sur le fichier *Sports\_enu.top* pour faire apparaître la fenêtre *Script editor*.



```
topic: ~Sports()
language: enu
```

```

topic: ~Sports()
language: frf
1: u:(Quel est ton sport préféré ?) Le football
2: u:(Quel est le sport le plus populaire à
3: Montreal?) Le Hockey sur glace|
4:
5:

```

Voici un exemple de règle :

**topic: ~Sports()** : La première ligne présente le sujet choisi.

**language: frf** : La deuxième ligne définit la langue.



Ces deux premières lignes sont inscrites automatiquement. Vous ne devez pas les modifier.

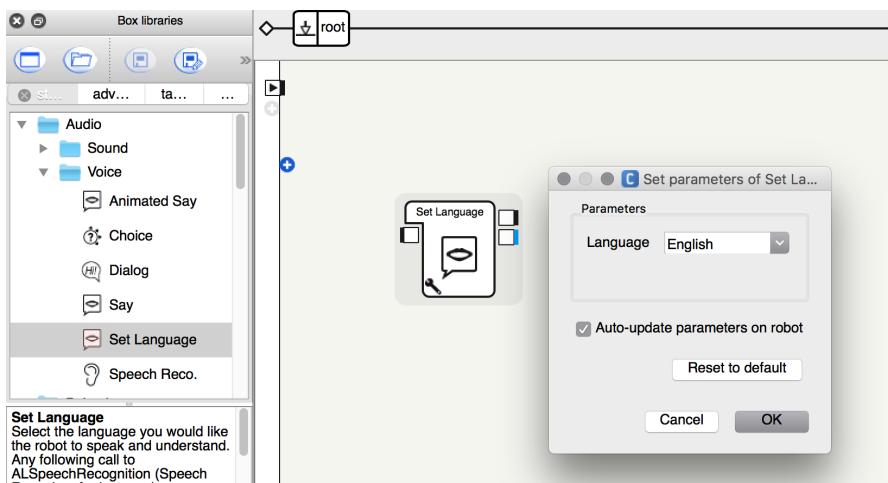
4. À présent, vous allez rédiger les règles.

- Vous devez commencer par écrire **u:**. S'il est bien tapé, il apparaît en bleu.
- Ensuite, vous écrivez entre parenthèses ce que NAO doit reconnaître vocalement, c'est-à-dire l'entrant humain. (**Quel est ton sport préféré ?**)
- Enfin, vous tapez la réponse du robot. **Le football**

Dans cet exemple, quand vous demanderez oralement à PEPPER « Quel est ton sport préféré ? », il vous répondra « Le football ».

5. Maintenant, vous devez ajouter dans votre espace de travail la boîte *Set Language* du sous-dossier VOICE du dossier AUDIO.

6. Ensuite, cliquez sur la clé pour définir la langue.



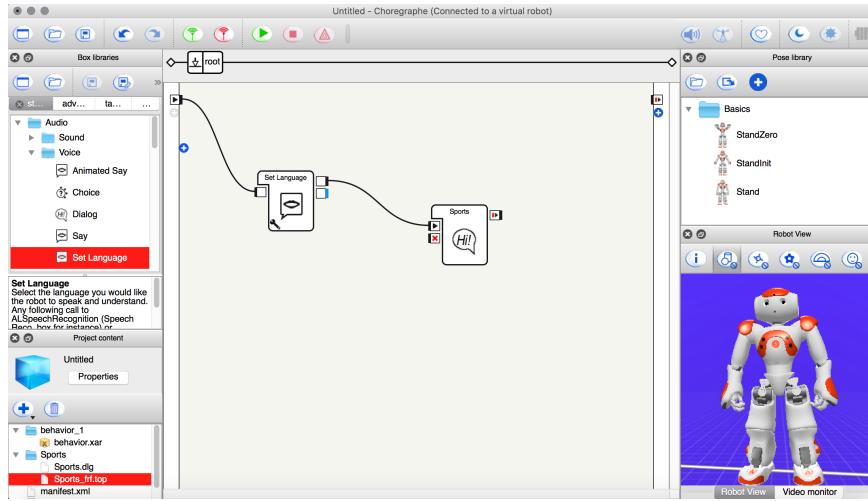
7. Vous devez lier l'input de cette boîte à l'onStart input et son output à la boîte *Dialog*.

8. Pour tester la séquence, cliquez sur le bouton lecture.

Les réponses du robot apparaissent dans la fenêtre *Robot View*.

9. Vous pouvez afficher la fenêtre *Dialog* et l'utiliser pour interagir avec le robot. Cliquez sur *View* dans la barre d'outils, puis appuyez sur *Dialog*.

Tapez des répliques de la boîte que vous avez créée et PEPPER vous répond. Cette fenêtre vous permet de visionner l'ensemble des échanges oraux.



Maintenant que vous êtes capable de créer des règles simples pour interagir avec Pepper via le script editor du mode *Dialog*, vous allez apprendre à les complexifier.

Pour enrichir vos scripts, vous pouvez ajouter des délimiteurs.

- Les crochets [ ] permettent de créer des choix. Les mots mis entre crochets dans l'entrée humaine peuvent se substituer, c'est-à-dire que tu peux employer l'un d'entre eux, et ceux dans la sortie du robot sont annoncés de façon séquentielle afin de créer des réponses variables.

```

topic: ~presentations()
language: frf
u: ([Bonjour salut bienvenue]NAO) [Enchanté
bonjour] je suis content de te rencontrer

```

Dans cet exemple, vous pouvez dire « Bonjour PEPPER », « Salut PEPPER » ou « Bienvenue PEPPER » pour que le robot réagisse. Le robot répondra « Enchanté, je suis content de te rencontrer » ou « Bonjour, je suis content de te rencontrer ».

- Les accolades {} permettent d'éditer des mots et des phrases optionnels dans une règle. L'humain peut utiliser les termes facultatifs tandis que le robot va répondre de manière séquentielle avec ou sans les mots optionnels.

```

1 topic: ~presentations()
2 language: frf
3 u: (Bonjour {NAO}) Enchanté {Nicolas}
4
5

```

Dans cet exemple, vous pouvez dire « Bonjour » ou « Bonjour PEPPER » pour qu'il réagisse. Le robot répondra « Enchanté » ou « Enchanté Nicolas ».



Dans un choix, une partie optionnelle ne peut être utilisée qu'à l'intérieur d'une phrase délimitée.

```

1 topic: ~presentations()
2 language: frf
3 u: ([ "Bonjour{NAO}" "salut {NAO}" ]) [ "Bonjour {Nicolas}" "Enchanté {Nicolas}" ]
4
5

```

- Les guillemets <> permettent de délimiter des phrases. Ainsi, vous pouvez placer une phrase au lieu d'un mot dans un choix ou une partie optionnelle.

```

1 topic: ~presentations()
2 language: frf
3 u: (["salut NAO" bonjour "bienvenue NAO"])
4 [Bonjour "Enchanté jeune humain"]

```

Dans cet exemple, vous pouvez dire « Salut PEPPER », « Bonjour » ou « Bienvenu PEPPER » pour qu'il réagisse. Il te répondra « Bonjour » ou « Enchanté jeune humain ».

Note : Pour modifier la langue de votre conversation cliquez sur *Sports.dlg*

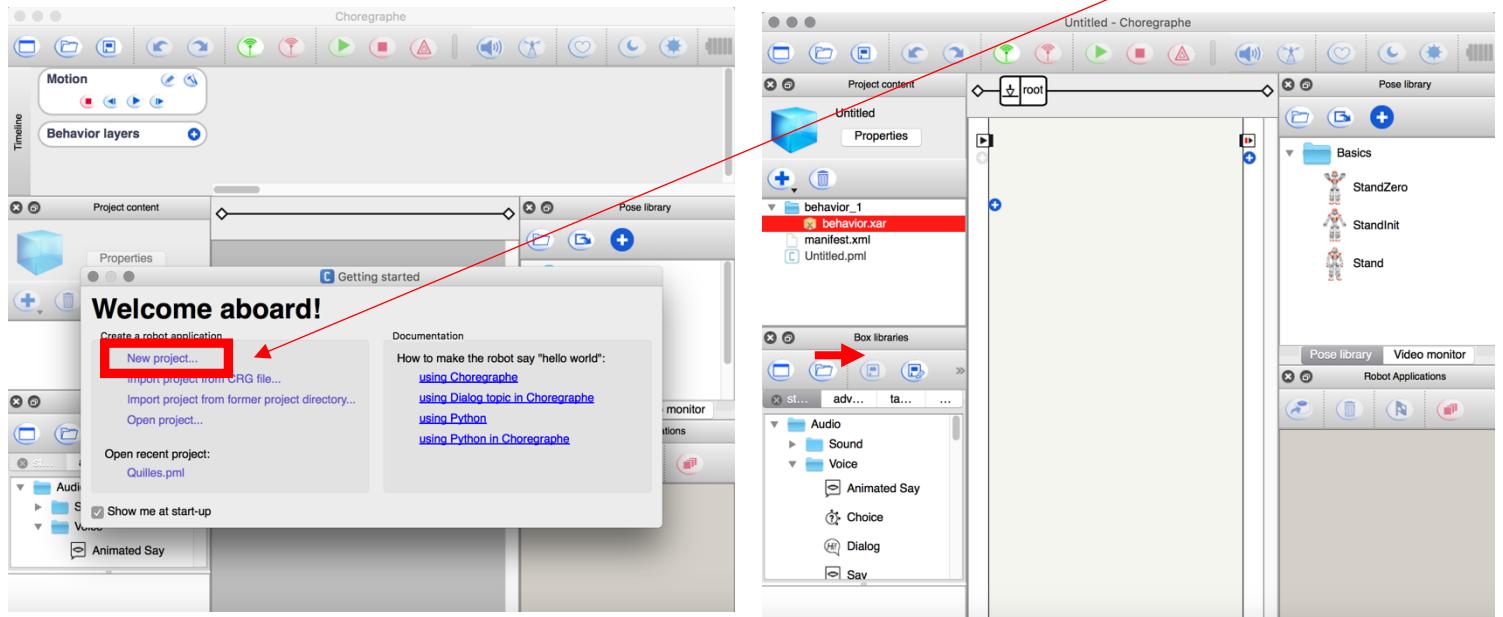
## 2.9 Création d'un comportement autonome

Grâce au logiciel *Choregraphe*, il est possible d'améliorer la vie autonome du robot en y installant des applications qui s'exécuteront dans le cadre *ALAutonomousLife*. Ces comportements peuvent être solitaires ou interactifs. Le comportement solitaire désigne une action où le robot réagit seul contrairement au comportement interactif qui nécessite une interaction humaine.

Exemple de comportement solitaire : Lorsque le robot ne détecte aucun visage dans les 30 secondes, il peut dire « Je me sens très seul ... ».

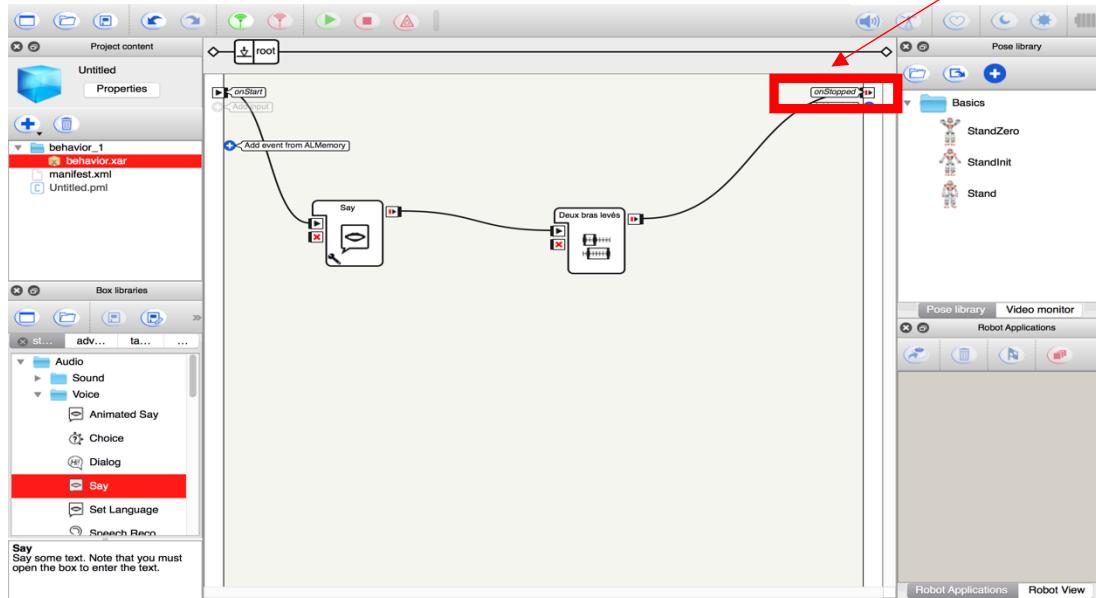
Exemple de comportement interactif : On demande au robot « Joue aux quilles ». Celui-ci répond « D'accord, place-moi devant les quilles et mets la balle dans ma main », puis il lance la balle.

1. La première étape est de créer un nouveau projet. Ouvrez *Choregraphe* et appuyez sur New project. Une nouvelle fenêtre de travail s'ouvre.

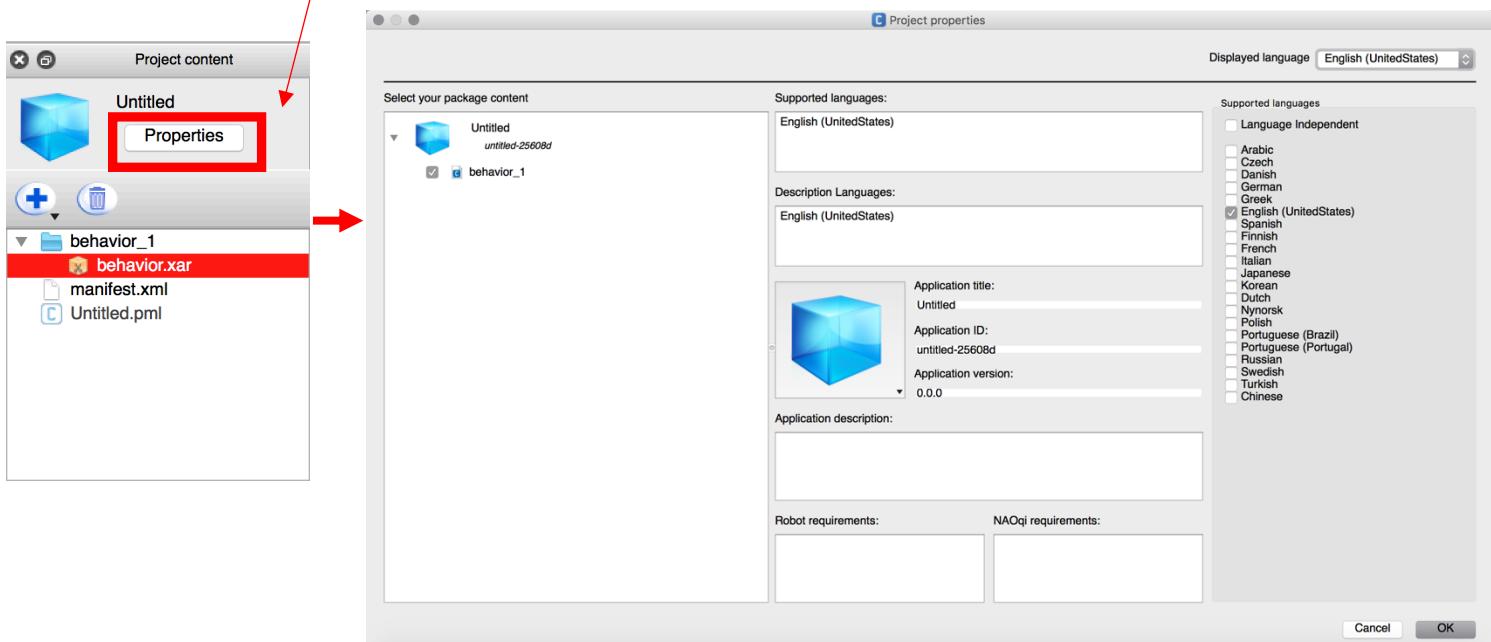


2. Créez un nouveau comportement. Par exemple : Dire « J'aime le chocolat » et lever les deux bras.

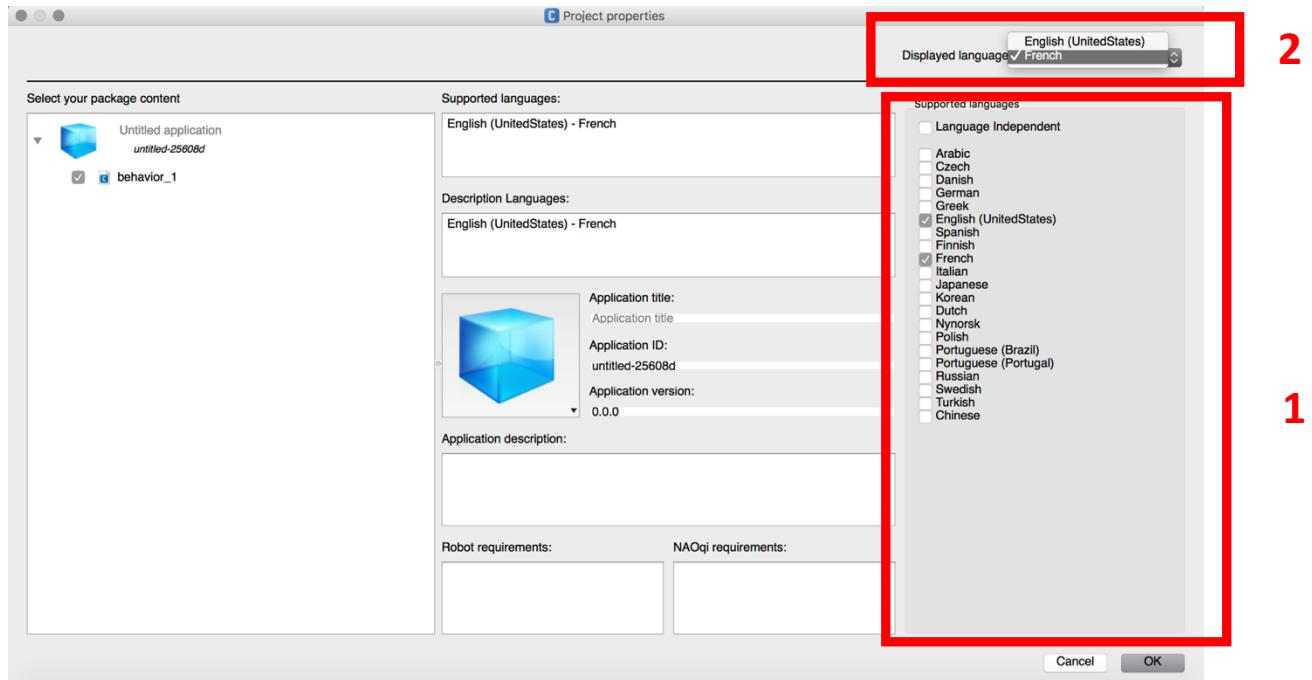
Pour une activité autonome, il est très important de relier le comportement à l'output onStopped pour qu'il se termine.



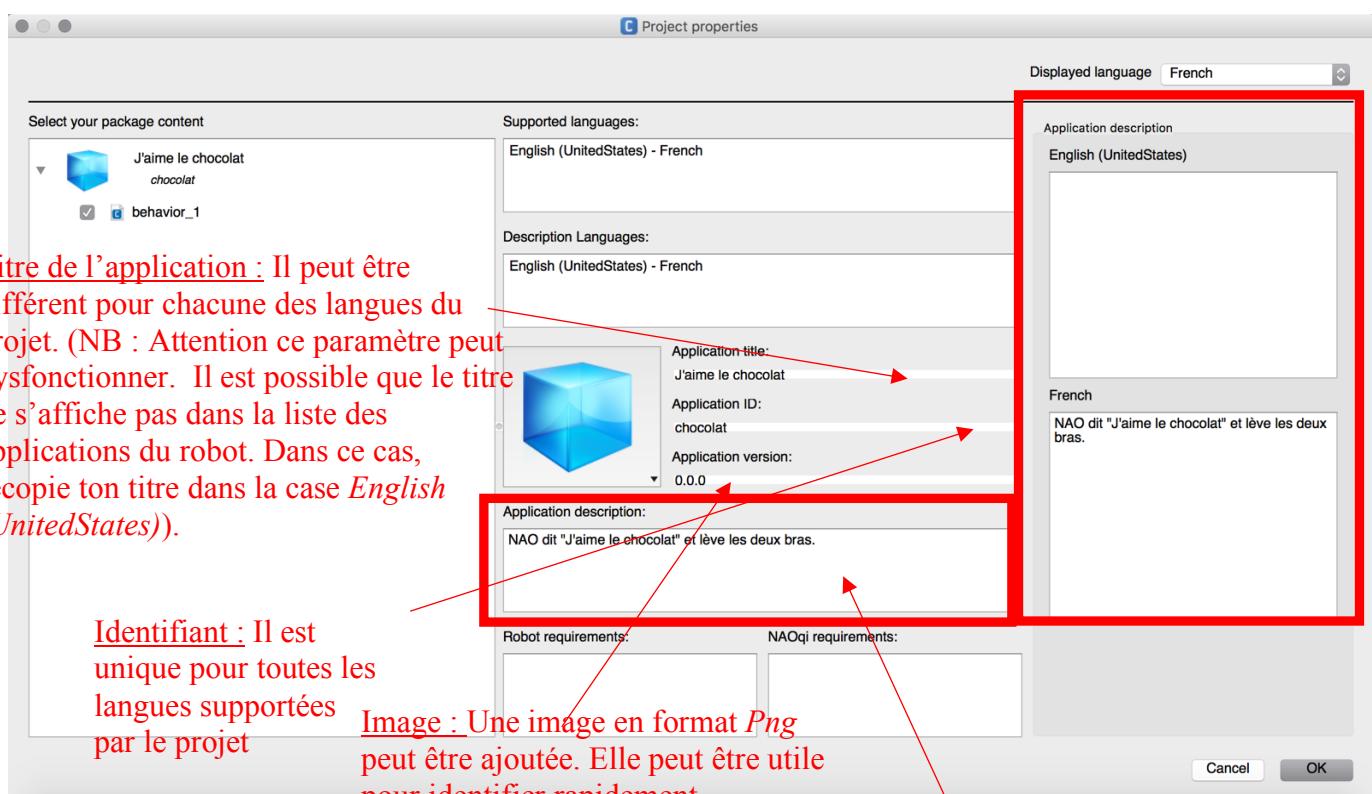
3. Maintenant vous allez devoir définir les propriétés du projet. Dans la fenêtre *Project content*, cliquez sur le bouton Properties. La fenêtre *Project properties* apparaît.



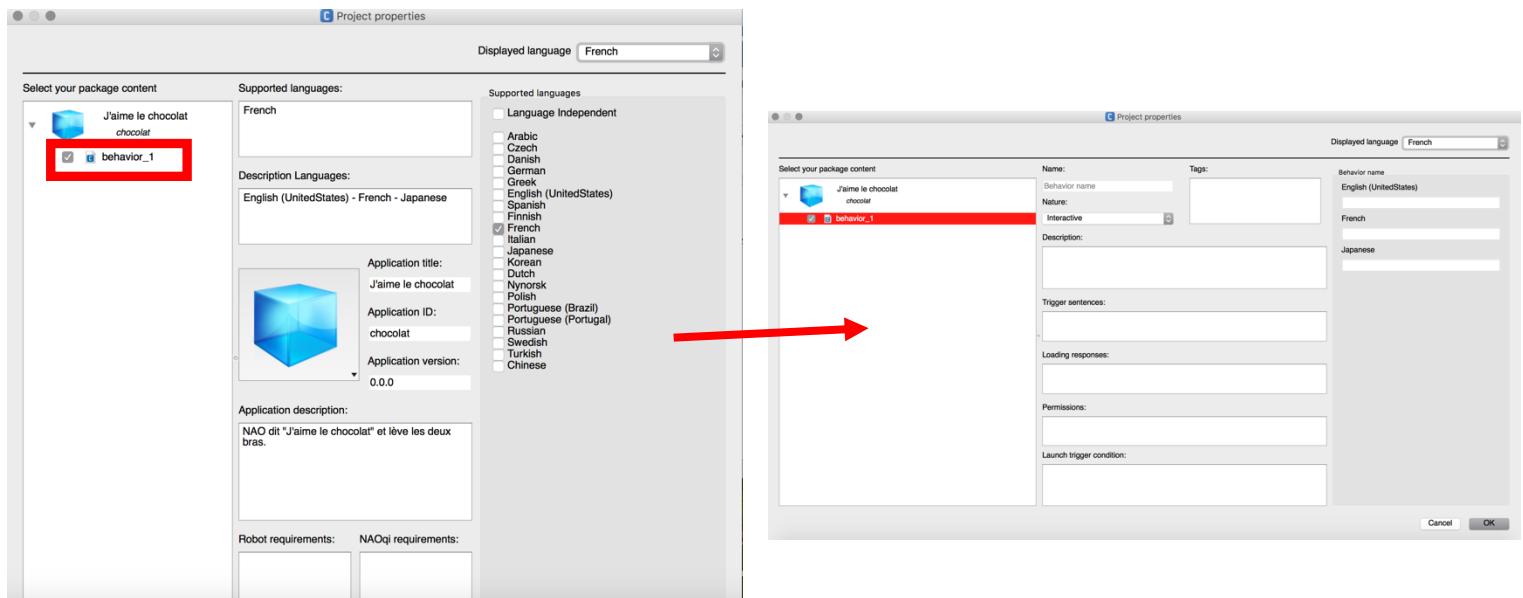
4. Définissez la langue de votre projet en cochant la case correspondante (1). Vous pouvez en choisir plusieurs. Puis choisissez la langue à afficher dans la fenêtre *Project properties* (2)



5. Donnez un nom à votre application et notez une description. Pour chacune de ces caractéristiques, cliquez sur le cadre correspondant pour l'éditer. Les détails de l'encadré sélectionné s'afficheront à droite de la fenêtre.



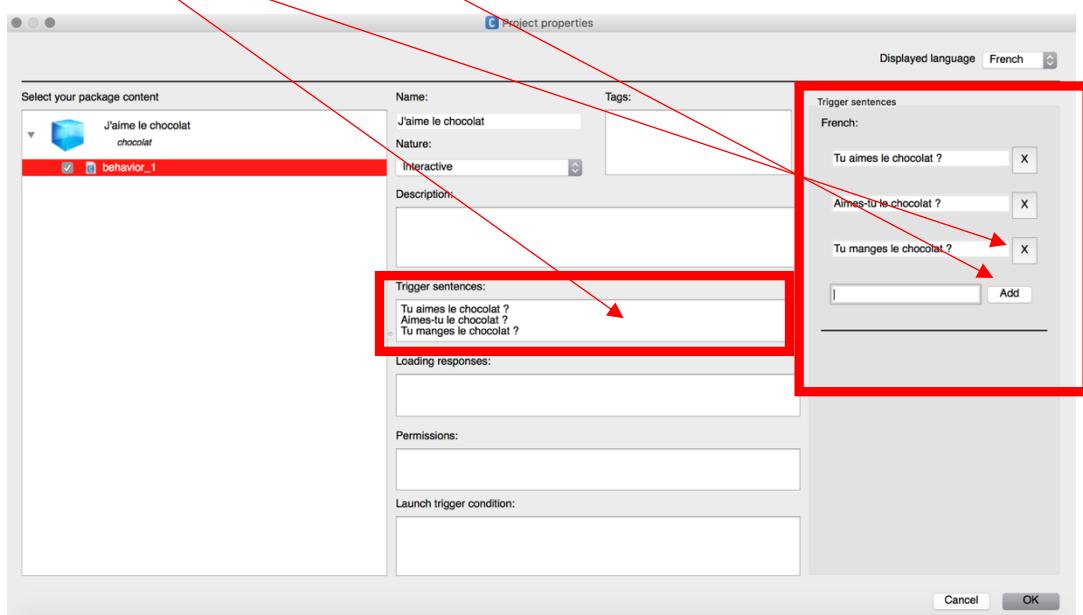
6. Désormais, vous allez définir les propriétés du comportement. Cliquez sur le nom du comportement dans la fenêtre *Select your package content*. Une nouvelle page apparaît.



7. Donnez un nom au comportement et choisissez sa nature (solitaire ou interactif).

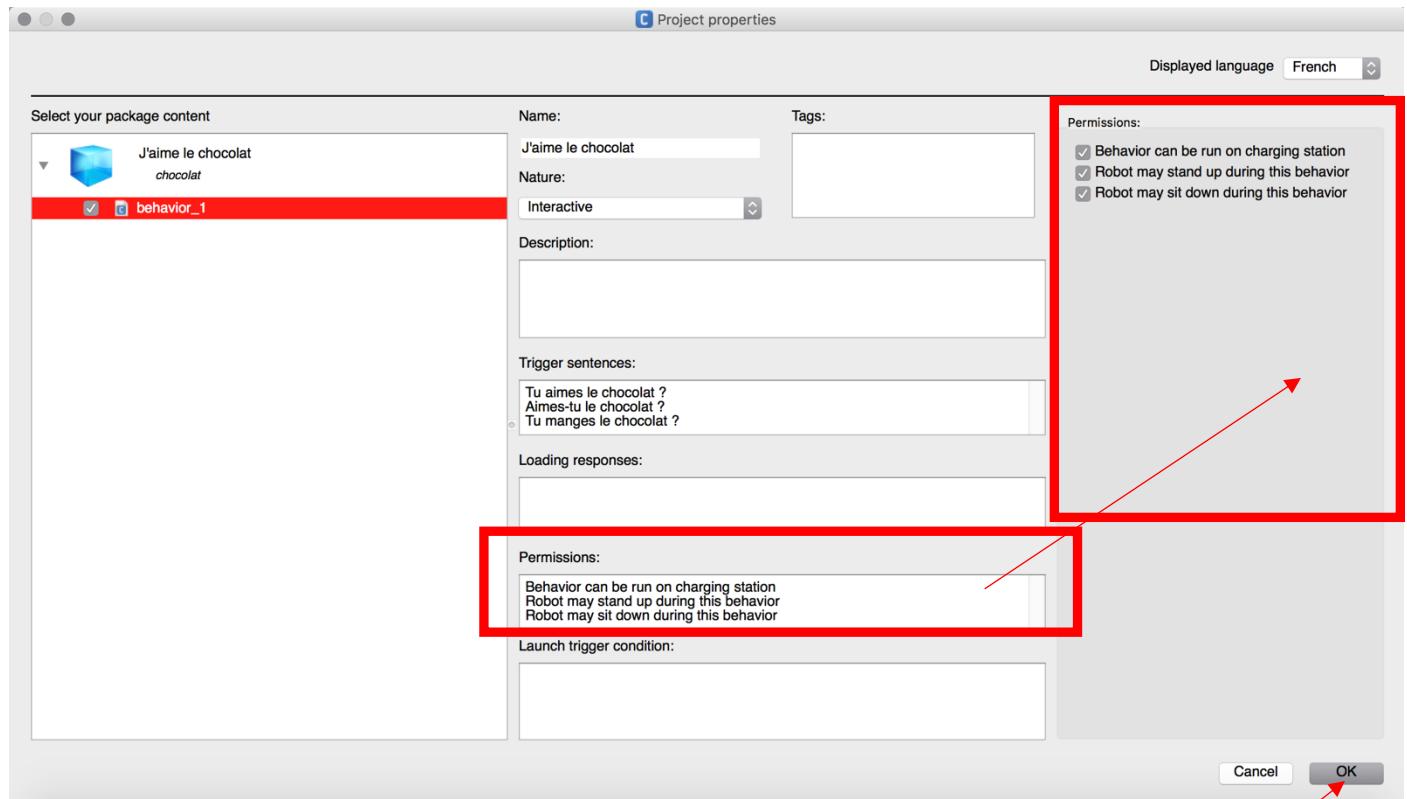


8. Inventez des phrases de déclenchement pour démarrer le comportement. Cliquez sur le cadre *Trigger sentences*. Les détails s'affichent à droite. Inscrivez une phrase dans la zone prévue à cet effet et cliquez sur le bouton *Add* pour l'ajouter au comportement. Vous pouvez la supprimer en cliquant sur *la croix*.



9. Le cadre *Loading responses* permet d'inscrire des phrases-réponses exprimées par PEPPE. Elles sont dites avant le lancement du comportement. Vous n'êtes pas obligé d'en mettre. Si vous laissez ce champ vide, des réactions automatiques seront exécutées durant le chargement du programme.

10. Définissez les permissions que vous souhaitez. Cliquez sur le cadre *Permissions*. Les détails s'affichent à droite. Sélectionnez des permissions pour votre programme en les cochant.



Le comportement peut être joué quand le robot est debout.

Le comportement peut être joué quand le robot est sur sa station de charge.

Le comportement peut être joué quand le robot est assis.

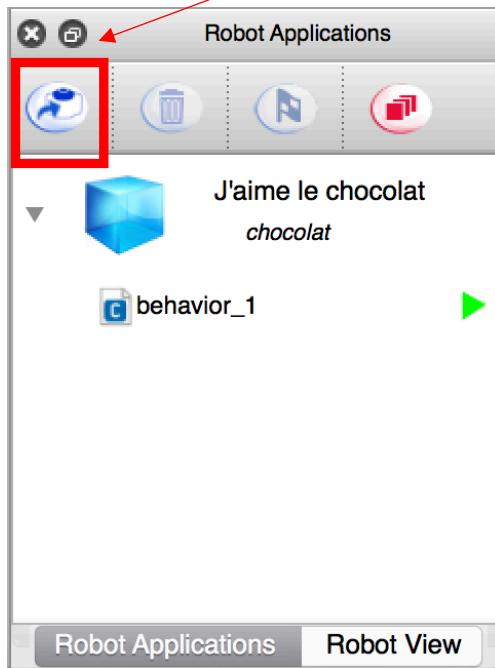
11. Le cadre *Launch trigger condition* donne la possibilité d'ajouter des conditions pour le lancement autonome de votre comportement. Il faut avoir de fortes connaissances en programmation pour utiliser cette fonctionnalité. Vous pouvez retrouver des informations dans la documentation d'Aldebaran :

<http://doc.aldebaran.com/2-1/naoqi/core/triggerconditions.html>

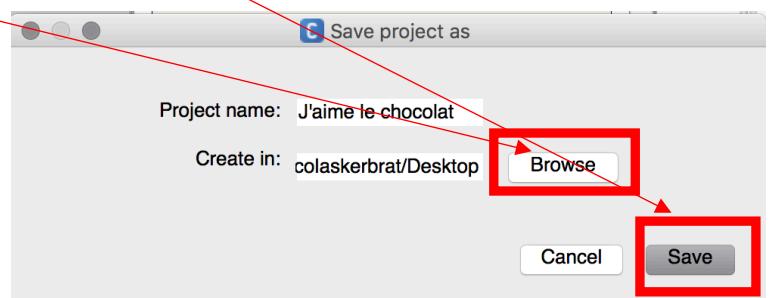
12. Appuyez sur le bouton **OK**.

13. Installez le programme dans la mémoire du robot en cliquant sur le bouton [Package and install current project to the robot](#) dans la fenêtre *Robot Applications*. Après quelques secondes, il apparaît dans la liste.

NB : Vous avez besoin d'un robot réel pour tester le comportement dans la vie autonome.



14 : Vous pouvez enregistrer le projet sur votre ordinateur pour lui apporter des modifications ultérieurement. Appuyez sur la [disquette](#). La fenêtre *Save project as* apparaît si c'est la première fois que vous enregistrez le projet. Nommez le fichier. Choisissez l'emplacement de sauvegarde en cliquant sur le bouton [Browse](#) et cliquez sur le bouton [Save](#). Si le projet est déjà enregistré, cela l'actualisera.



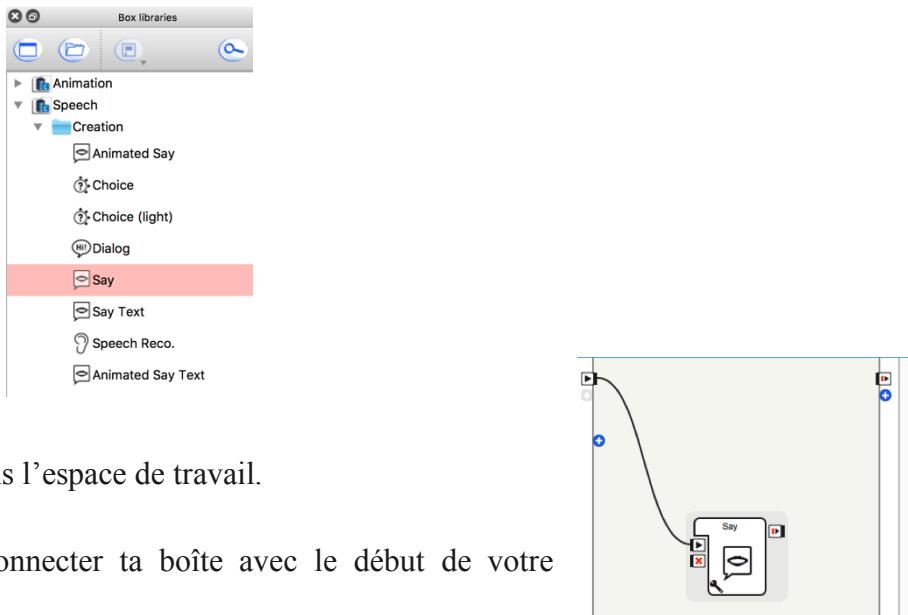
### 3. Les fonctionnalités de PEPPER

#### 3.1 Parler

Par défaut, Pepper est réactif et capable de dialoguer avec toi, grâce à sa vie autonome. Vous pouvez retrouver les explications et les dialogues pré-enregistrés de la page 21 à la page 27. Vous pouvez aussi le faire parler en le programmant avec Choregraphe. Les boîtes *Say* et *Animated Say* sont les plus utilisées.

##### 3.1.1 Boîte *Say*

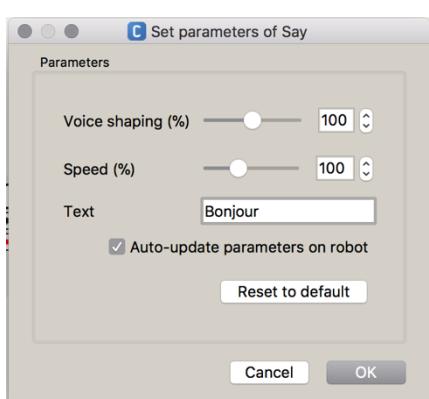
1. Dans la fenêtre *Box libraries*, cherchez le dossier SPEECH, puis CREATION :



2. Déplacez la boîte *Say* dans l'espace de travail.

3. Vous pouvez ensuite connecter ta boîte avec le début de votre séquence :

4. Cliquez sur la clé à molette de la boîte *Say* pour faire apparaître la fenêtre *Set parameters of Say*.



- Inscrivez ce que vous voulez que PEPPER dise dans l'onglet *Text*.

- L'onglet *Voice shaping* permet de modifier la voix du robot. Plus le nombre est faible et plus le son est grave. Inversement plus le nombre est élevé et plus le son est aigu.

- L'onglet *Speed* permet de contrôler la vitesse de parole.



5. Pour tester ta séquence, appuyez sur :

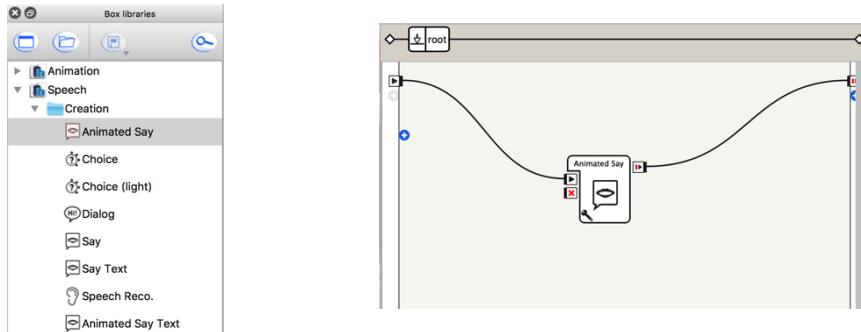


ATTENTION : Vous devez tester la séquence avec le vrai robot pour entendre la voix.

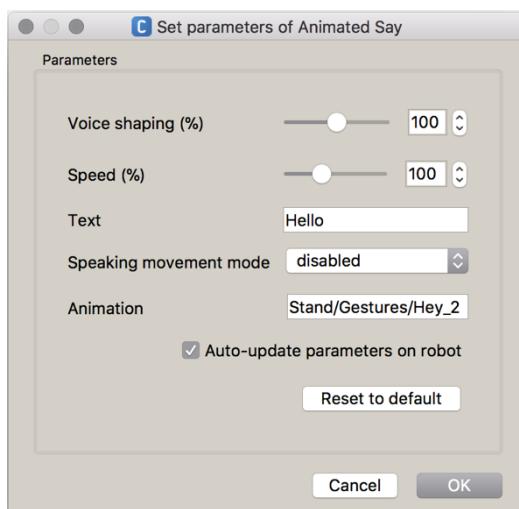
### 3.1.2 Boîte Animated Say

La boîte *Animated Say* est programmé pour que PEPPER dise bonjour en parlant et en bougeant les bras et le corps simultanément.

1. Prenez la boîte *Animated Say* dans le sous-dossier VOICE du dossier AUDIO et glissez-la dans l'espace de travail. Connectez la boîte aux entrées et sorties.



2. Cliquez sur la clé à molette de la boîte *Animated Say* pour faire apparaître la fenêtre *Set parameters of Animated Say*.

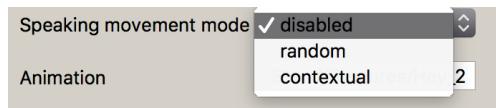


- Inscrivez ce que vous voulez que PEPPER dise dans l'onglet *Text*.

- L'onglet *Voice shaping* permet de modifier la voix du robot. Plus le nombre est faible et plus le son est grave. Inversement plus le nombre est élevé et plus le son est aigu.

- L'onglet *Speed* permet de contrôler la vitesse de parole.

- L'onglet *Speaking movement mode* permet de choisir un des trois modes de mouvement.



- *Disabled* permet de jouer le mouvement que tu as inséré dans l'onglet *Animation* ;
  - *Random* permet de jouer des mouvements au hasard ;
  - *Contextual* permet de jouer des mouvements en contexte avec ce que le robot dit.
- L'onglet *Animation* permet d'insérer le mouvement de votre choix. Vous retrouvez la bibliothèque des postures dans la bibliothèque d'Aldebaran : <http://doc.aldebaran.com/2-5/naoqi/motion/alanimationplayer-advanced.html>

3. Cliquez sur *Ok* pour revenir à votre espace de travail.



4. Pour tester votre séquence, appuyez sur :



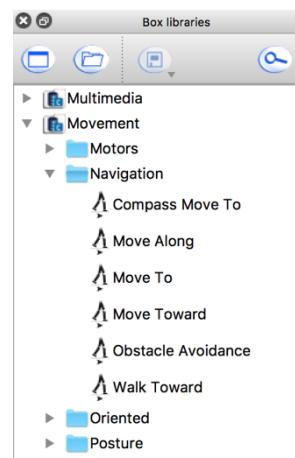
ATTENTION : Vous devez tester la séquence avec le vrai robot pour entendre la voix.

## 3.2 Se déplacer

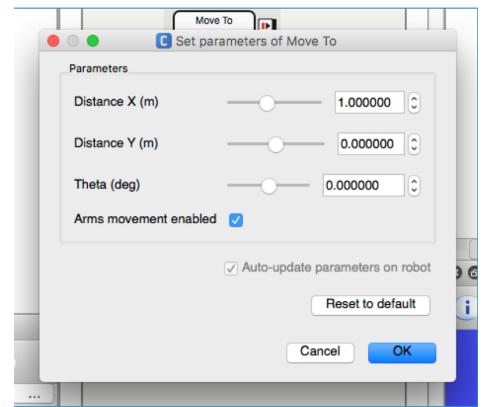
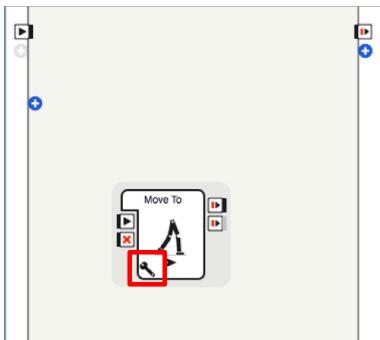
Plusieurs boîtes permettent à PEPPER de se déplacer. La boîte *Move to* est la plus utilisée. Pour qu'il se déplace, il doit être dans un endroit dégagé et le capot de charge doit être fermé.

### 3.2.1 *Move To*

1. Dans la fenêtre *Box libraries*, sélectionnez le dossier **MOVEMENT**, puis le sous-dossier **NAVIGATION** et repérez l'action *Move To* :



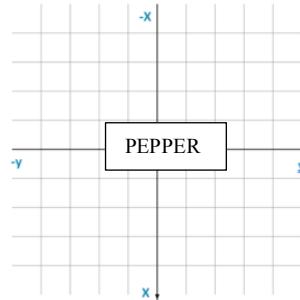
2. Appuyez sur *Move To* et déplacez la boîte dans l'espace de travail (fenêtre du centre).



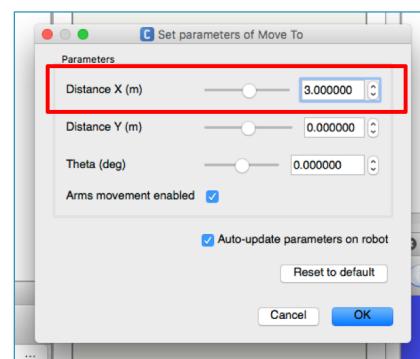
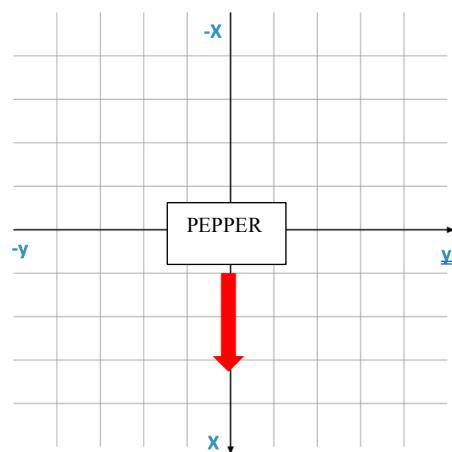
3. Appuyez ensuite sur pour modifier les paramètres de l'action. La fenêtre *Set parameters of Move To* va apparaître.

Imaginez votre robot PEPPER au centre de deux axes (X et Y).

Vue de haut

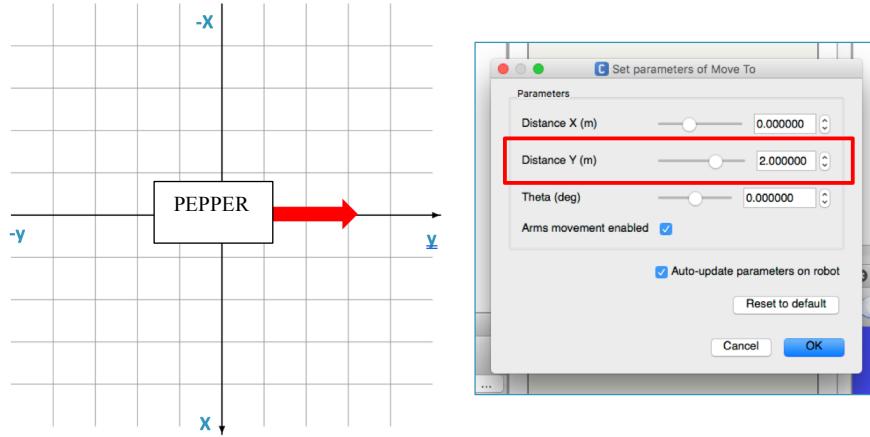


Pour faire avancer votre robot en avant, augmentez la valeur de X ( $X > 0$  et  $Y = 0$ ) :



Dans l'exemple, le robot PEPPER va avancer de 3 mètres vers l'avant, puisque  $X=3,000000$ .

Pour le déplacer vers la gauche, augmentez la valeur de Y ( $Y > 0$  et  $X = 0$ ) :

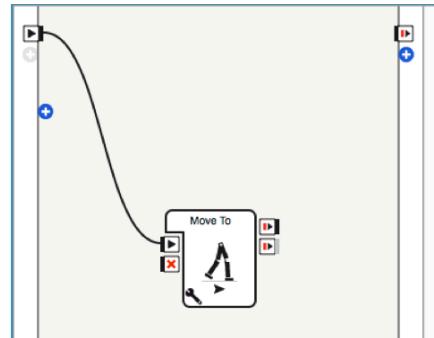


Dans l'exemple, mon robot va se déplacer à gauche de 2 mètres, puisque  $Y = 2,000000$ .



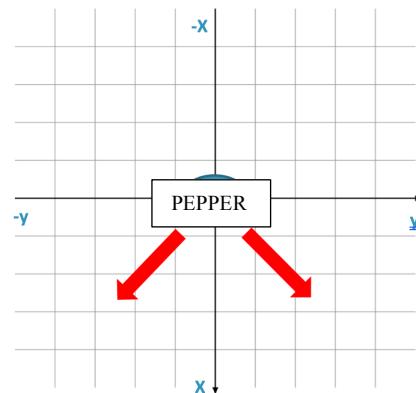
ATTENTION :

N'oubliez pas de faire votre lien avec *onStart*.



\*NOTE : Vous devez utiliser l'output gris pour que le signal soit envoyé malgré que le robot se soit arrêté avant la destination finale.

- Si les valeurs de X et de Y ont une valeur autre que 0 en même temps, votre robot avancera en diagonale.



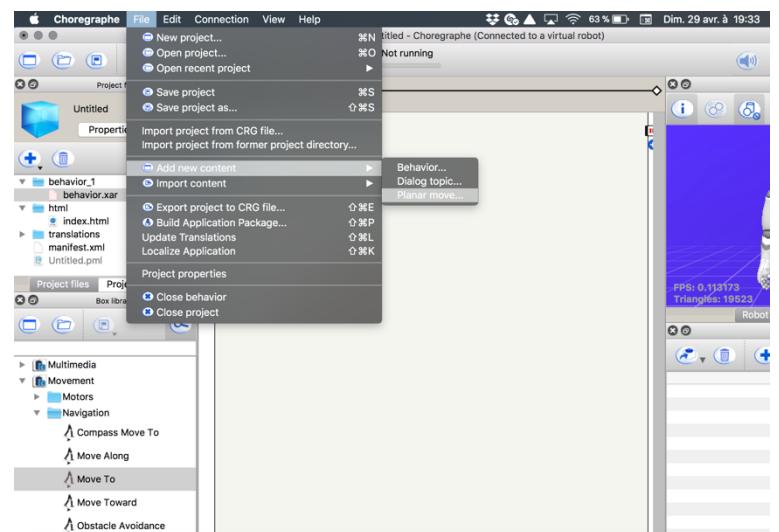
- Pour le faire reculer ou pour le déplacer vers la droite, vous devez utiliser les valeurs négatives de X et Y.

Par exemple : si vous voulez qu'il recule de 2 mètres, X = -2,000000 ; si vous voulez qu'il avance vers la droite de 3 mètres, Y = -3,000000

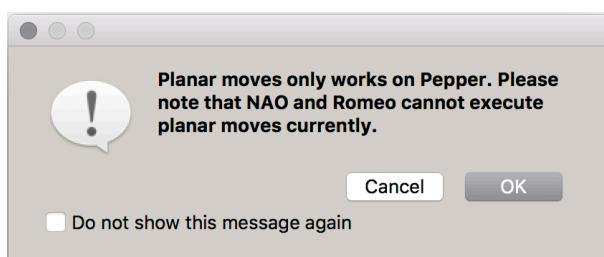
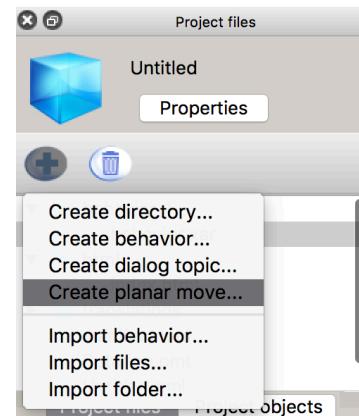
### 3.2.2 Move Along

La boîte *Move Along* permet au robot de se déplacer selon des trajectoires prédéfinies. Il faut donc créer des fichiers qui décrivent les mouvements du robot dans un espace plan. L'extension des fichiers est PMT.

1. Cliquez sur *File > Add new content > Planar move*

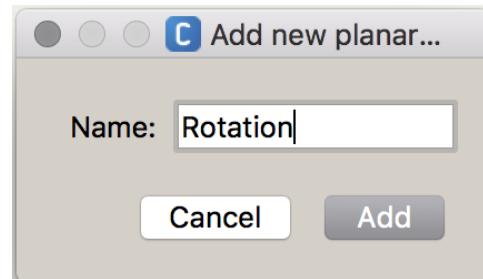


- ou alors cliquez sur le + > *Create planar move*

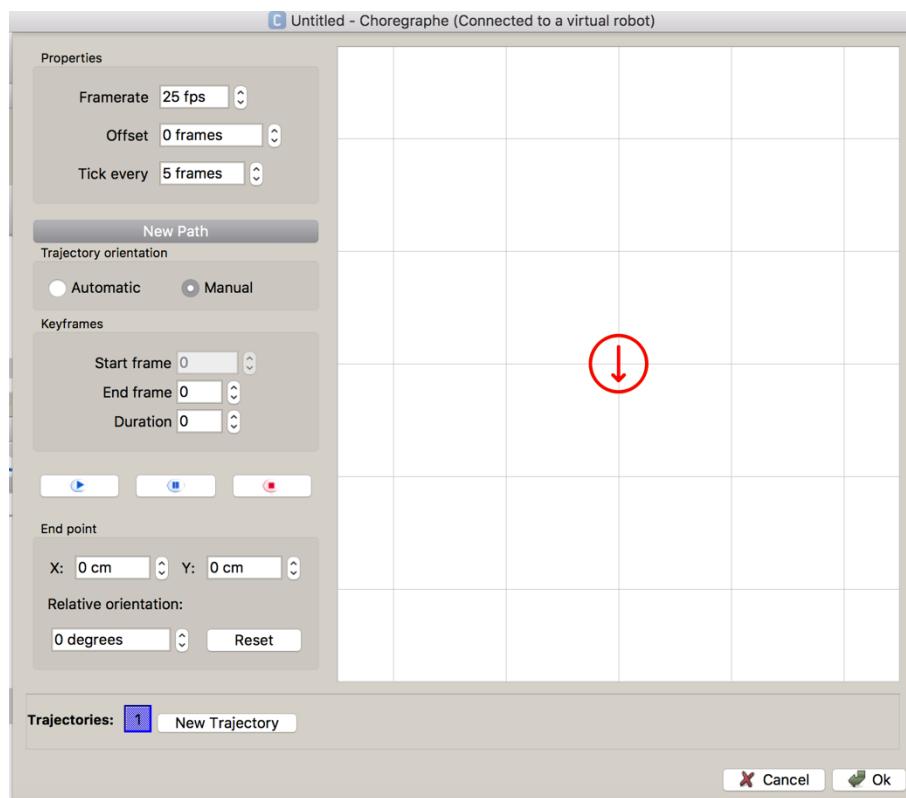


Un message d'information s'affiche. Cliquez sur *Ok*.

2. Choisissez un nom pour votre fichier.



3. Cliquez sur le bouton *Add* et créez votre déplacement planaire.



A : Cliquez sur la flèche rouge et déplacez la.

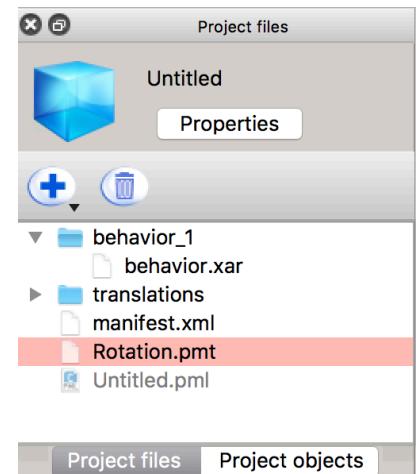
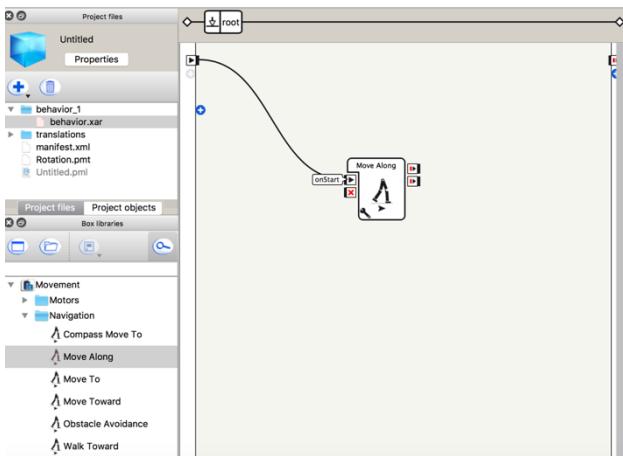
B : Faites défiler la roulette de votre souris ou le trackpad de votre ordinateur pour changer l'orientation de votre robot.

C : Appuyez sur la touche *Entrée* de votre clavier ou cliquez sur *New Path* pour enregistrer une position intermédiaire sur la trajectoire. Les déplacements suivants seront des arcs de cercle pour assurer un mouvement continu.

D : Continuez de créer des mouvements sur cette trajectoire en renouvellement les opérations précédentes ou alors cliquez sur *New trajectory* pour en créer une nouvelle.

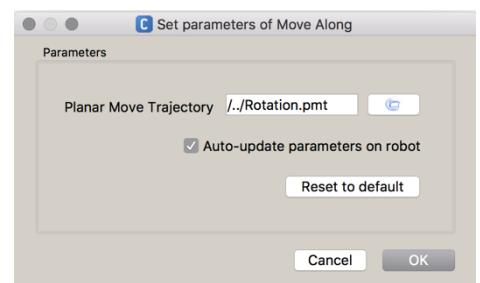
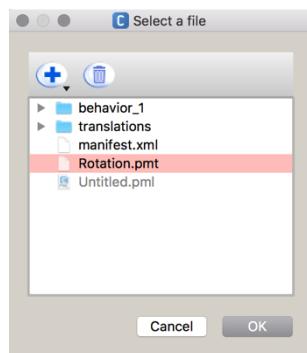
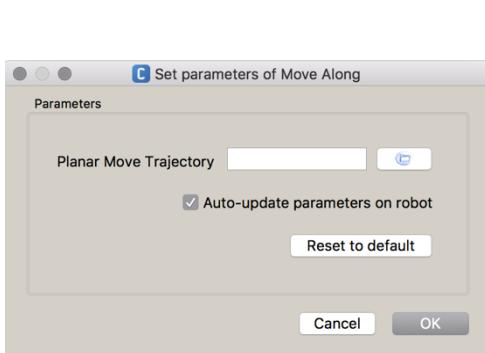
<p><b>Properties</b></p> <p>Framerate 25 fps</p> <p>Offset 0 frames</p> <p>Tick every 5 frames</p>	<p><i>Framerate et Tick</i> modifient l'affichage des trajectoires.</p> <p><i>Offset</i> retarde les trajectoires.</p>
<p>Trajectory orientation</p> <p><input checked="" type="radio"/> Automatic    <input type="radio"/> Manual</p>	<p>La fonction automatique permet de définir automatiquement les orientations de Pepper.</p> <p>La fonction manuelle vous donne la possibilité de les définir vous-même.</p>
<p><b>Keyframes</b></p> <p>Start frame 253</p> <p>End frame 978</p> <p>Duration 725</p>	<p>La fonction durée permet d'affiner le temps nécessaire à l'exécution d'une trajectoire.</p>
<p>▶   ▶■   ■○</p>	<p>Ces boutons permettent de visualiser la trajectoire.</p>
<p>End point</p> <p>X: -49 cm   Y: -31 cm</p> <p>Relative orientation:</p> <p>84 degrees   Reset</p>	<p>Vous pouvez affiner la position finale de la trajectoire.</p>
<p><b>Trajectories:</b> 1 2   New Trajectory</p>	<p>Vous pouvez sélectionner une trajectoire ou en créer une nouvelle.</p>

4. Cliquez sur *OK* quand vous avez terminé. Votre fichier apparaît dans la fenêtre *Project Files* dans le dossier TRANSLATIONS.

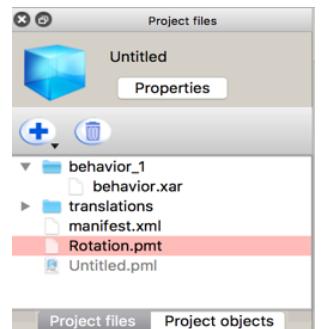


5. Glissez la boîte *Move Along* du sous-dossier NAVIGATION, du dossier MOVEMENT, dans votre espace de travail. Connectez-la à l'*onStart*.

6. Cliquez sur la clé à molette. La fenêtre *Set parameter of Move Along* apparaît. Cliquez sur l'onglet *Dossier* et sélectionnez le fichier de déplacement planaire de votre choix en format pmt. Cliquez sur *OK*.



\*Vous pouvez à tout moment modifier votre déplacement planaire en double-cliquant sur votre fichier dans la fenêtre *Project files*.

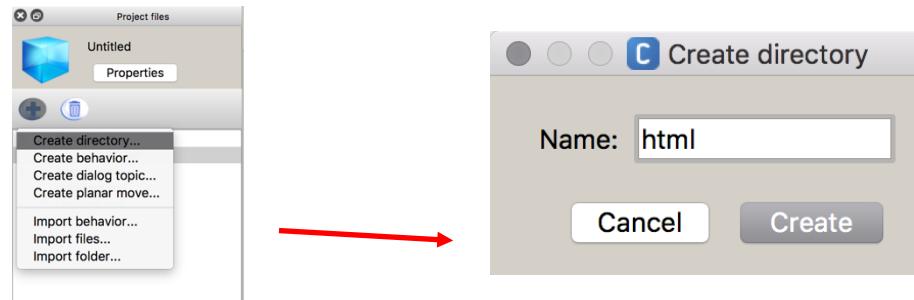


### 3.3 Utilisation de la tablette

#### 3.3.1 Afficher une image

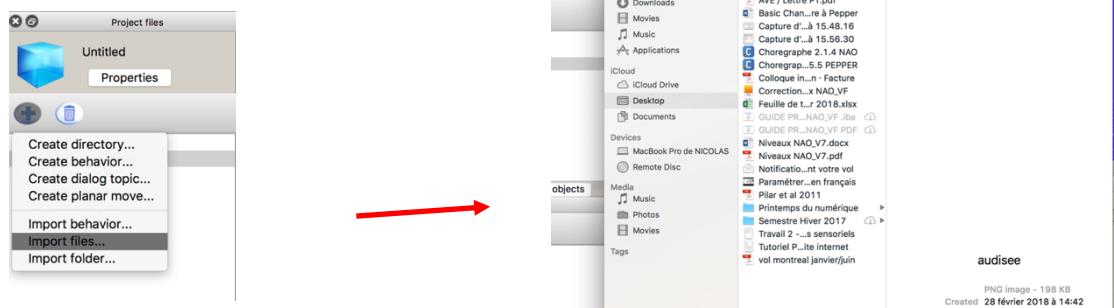
1. Créez un dossier html dans *Project files*.

Pour cela cliquez sur le +, puis sur *Create directory* et nommez le dossier *html*.

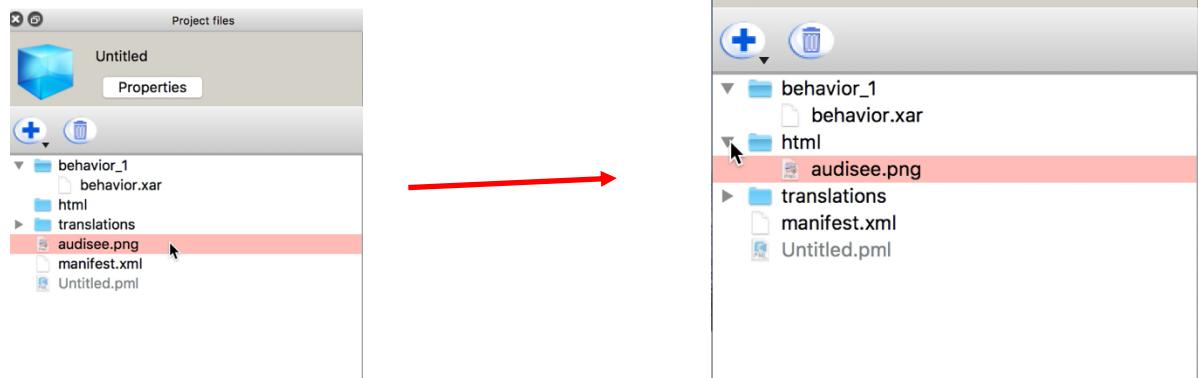


2. Ajoutez votre image dans le projet.

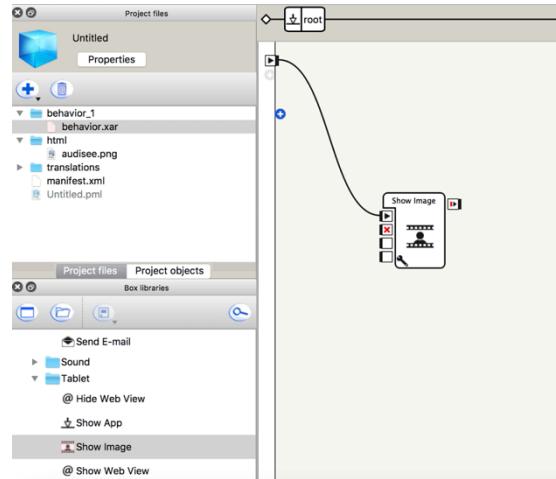
Pour cela cliquez sur le +, puis sur *import files*. Double-cliquez sur le fichier de votre choix en format png.



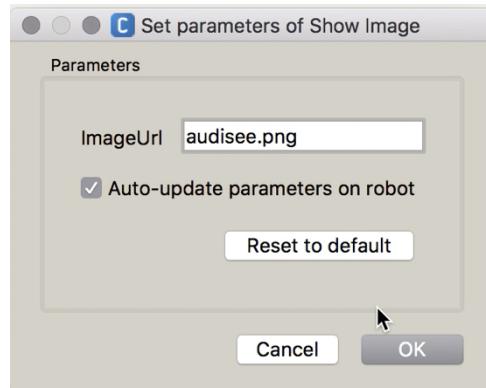
Il apparaît en bas dans la fenêtre *Project files*. Déplacez le dans le dossier *html*.



3. Glissez la boîte *Show Image*, du sous-dossier TABLET, du dossier MULTIMEDIA, dans ton espace de travail et connectez la à l' *onStart*.



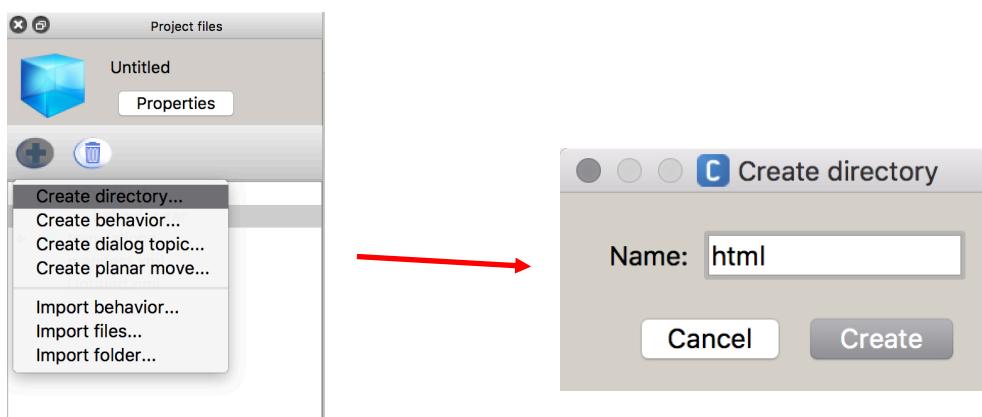
4. Cliquez sur la clé à molette. La fenêtre *Set parameters of Show Image* apparaît. Inscrivez le nom du fichier image et cliquez sur *OK*.



### 3.3.2 Afficher une vidéo

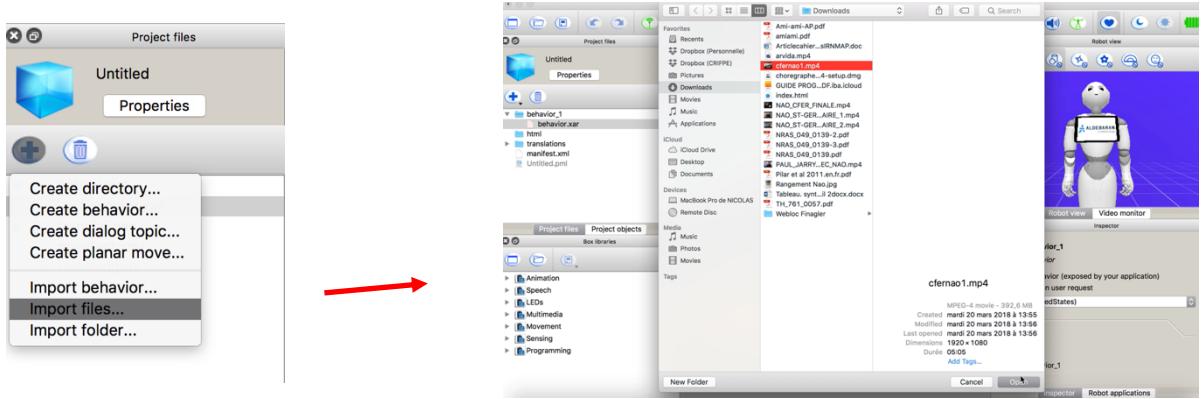
1. Créez un dossier html dans *Project files* .

Pour cela cliquez sur le +, puis sur *Create directory* et nommez le dossier *html*

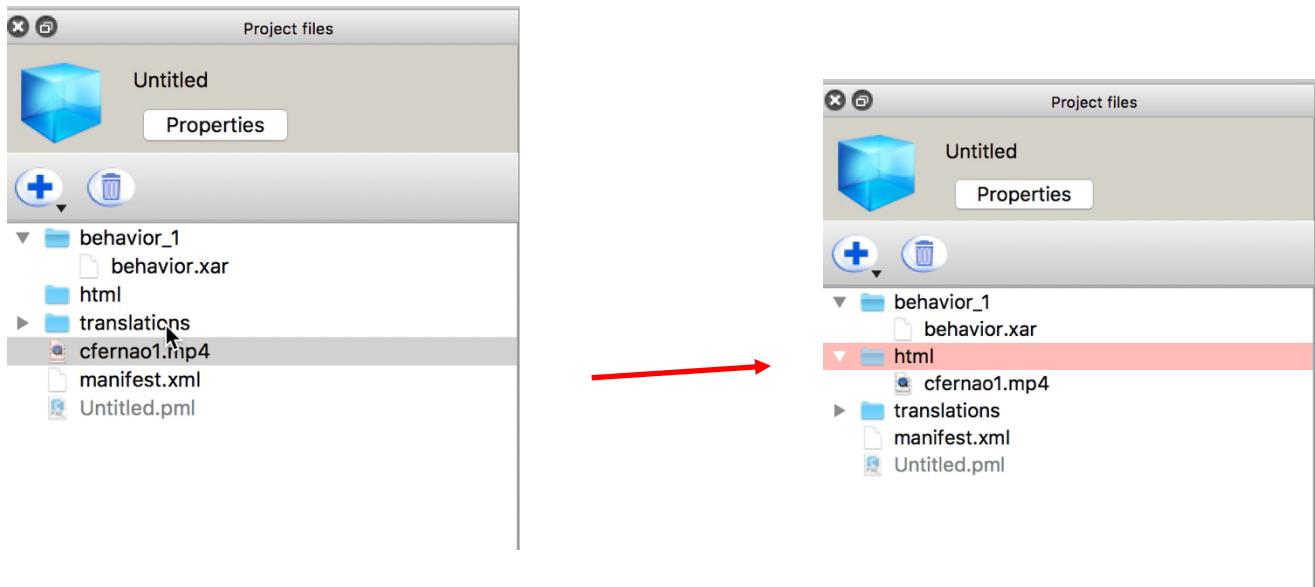


## 2. Ajoutez votre vidéo dans le projet.

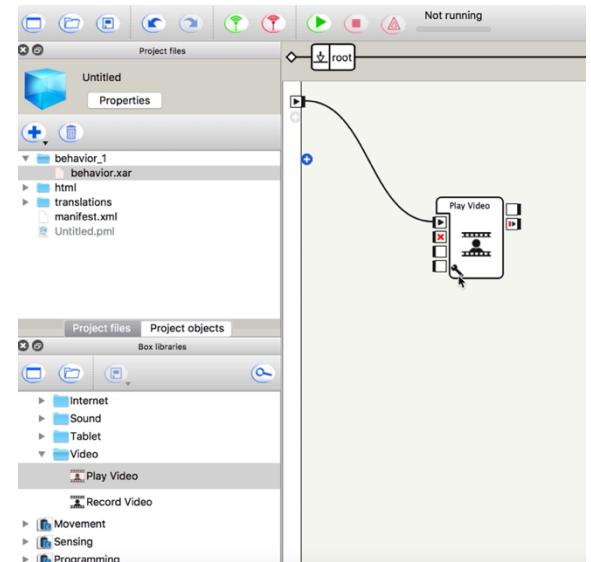
Pour cela cliquez sur le +, puis sur *import files*. Double-cliquez sur le fichier de votre choix en format mp4.



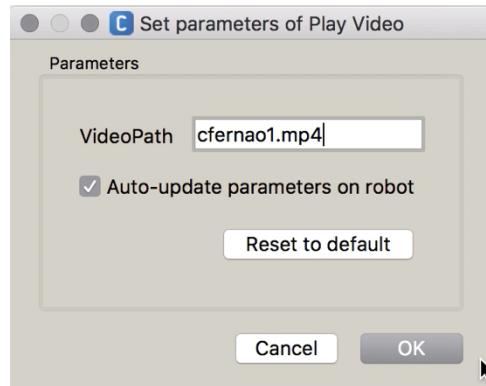
Il apparaît en bas dans la fenêtre *Project files*. Déplacez le dans le dossier *html*.



3. Glissez la boîte *Play Video*, du sous-dossier VIDEO du dossier MULTIMEDIA, dans votre espace de travail et connectez la à l' *onStart*.



4. Cliquez sur la clé à molette. La fenêtre *Set parameters of Play Video* apparaît. Inscrivez le nom du fichier vidéo.



### 3.3.3 Afficher un fichier html

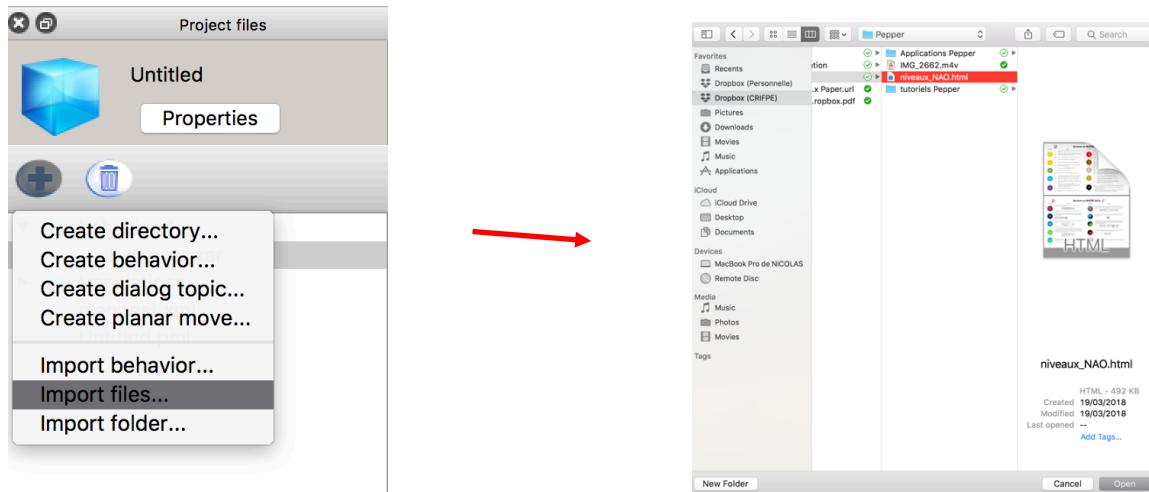
1. Créez un dossier html dans *Project files* .

Pour cela cliquez sur le +, puis sur *Create directory* et nommez le dossier *html*

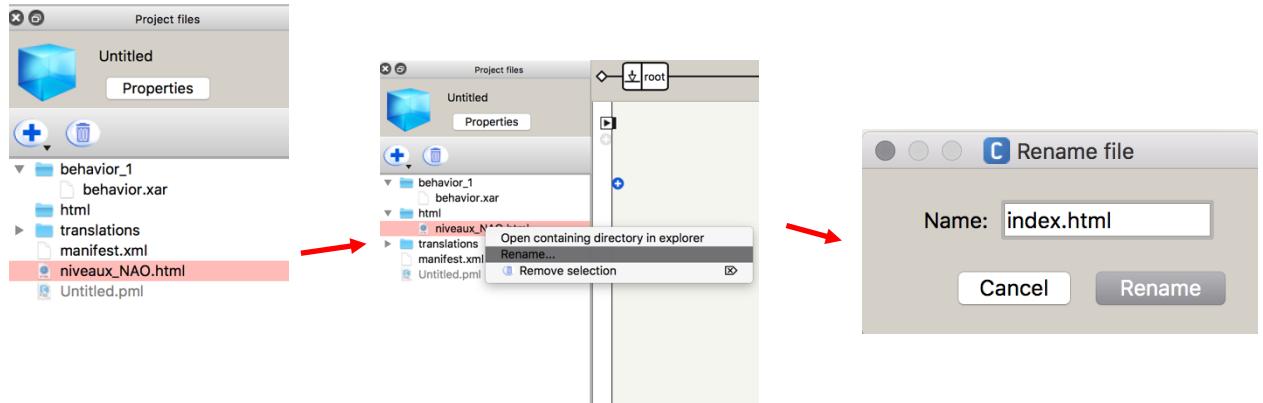


## 2. Ajoutez votre fichier en le renommant *index.html*

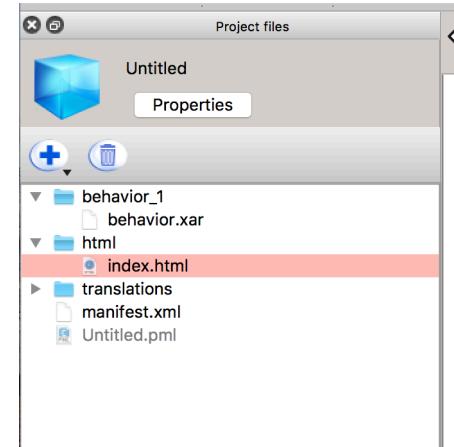
Pour cela cliquez sur le +, puis sur *import files*. Double-cliquez sur le fichier de votre choix en format html.

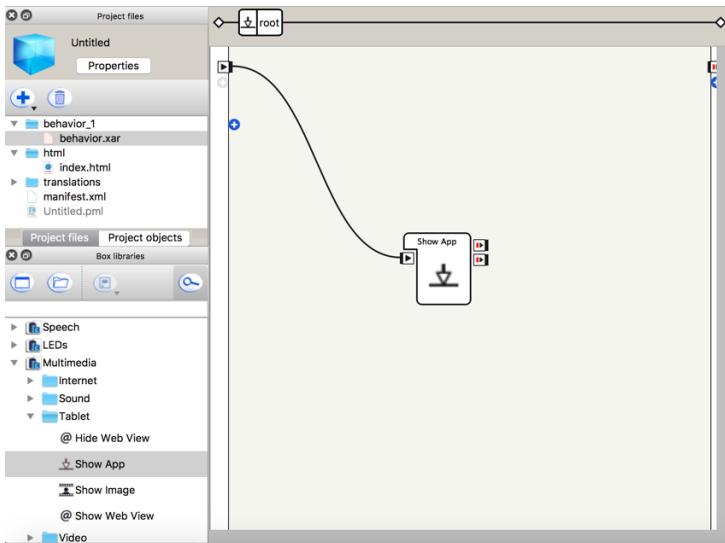


Il apparaît en bas dans la fenêtre *Project files*. Déplacez le dans le dossier *html* et renommez le *index.html*. Cliquez droit dessus et appuyez sur *Rename...*



Votre fenêtre *Project files* doit ressembler à celle de cette image :





3. Glissez la boîte *Show App*, du sous-dossier TABLET du dossier MULTIMEDIA, dans votre espace de travail et connectez la à l' *onStart*.

### 3.3.4 Afficher un site internet

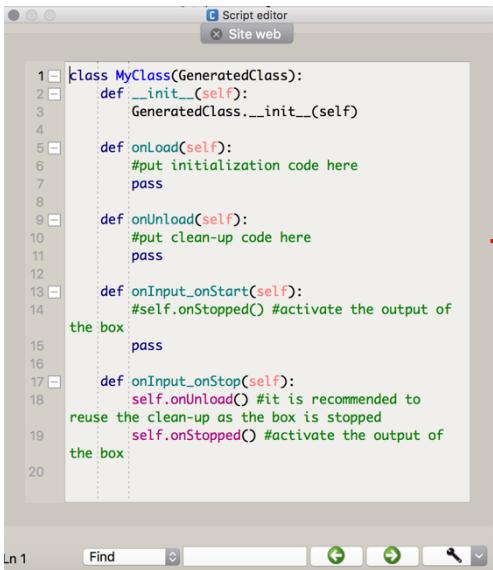
#### 1. Créez une boîte *Python*.

Cliquez droit dans votre espace de travail. Appuyez sur *Create a new box*, puis sur *Python*. La fenêtre *Edit box* apparaît. Vous pouvez nommer la boîte, lui donner une description et choisir une image.



## 2. Modifiez le code source.

Double-cliquez sur la boîte pour accéder au *Script editor*.



A screenshot of a Mac OS X-style application window titled "Script editor". The title bar also includes "karsenti.ca". The main pane shows the same code as the left window, but with several changes highlighted in green. Lines 14, 15, 16, 17, 18, 19, 20, 21, and 22 have been removed, indicating they were deleted. A red arrow points from the text "the box" in the original code towards the right-hand side of the image, where a comparison is shown.

```
1 class MyClass(GeneratedClass):
2     def __init__(self):
3         GeneratedClass.__init__(self)
4         self.web = ALProxy ("ALTabletService")
5
6     def onLoad(self):
7         #put initialization code here
8         pass
9
10    def onUnload(self):
11        #put clean-up code here
12        pass
13
14    def onInput_onStart(self):
15        self.web.enableWifi()
16        self.web.loadUrl ("http://www.karsenti.ca/code/projet-nao-en-education/")
17        self.web.showWebview()
18        self.onStopped()
19        pass
20
21    def onInput_onStop(self):
22        self.onUnload() #it is recommended to reuse the clean-up as the
23        #box is stopped
24        self.onStopped() #activate the output of the box
```

Voilà le nouveau script :

```
class MyClass(GeneratedClass):
    def __init__(self):GeneratedClass.__init__(self)
    self.web = ALProxy ("ALTabletService")

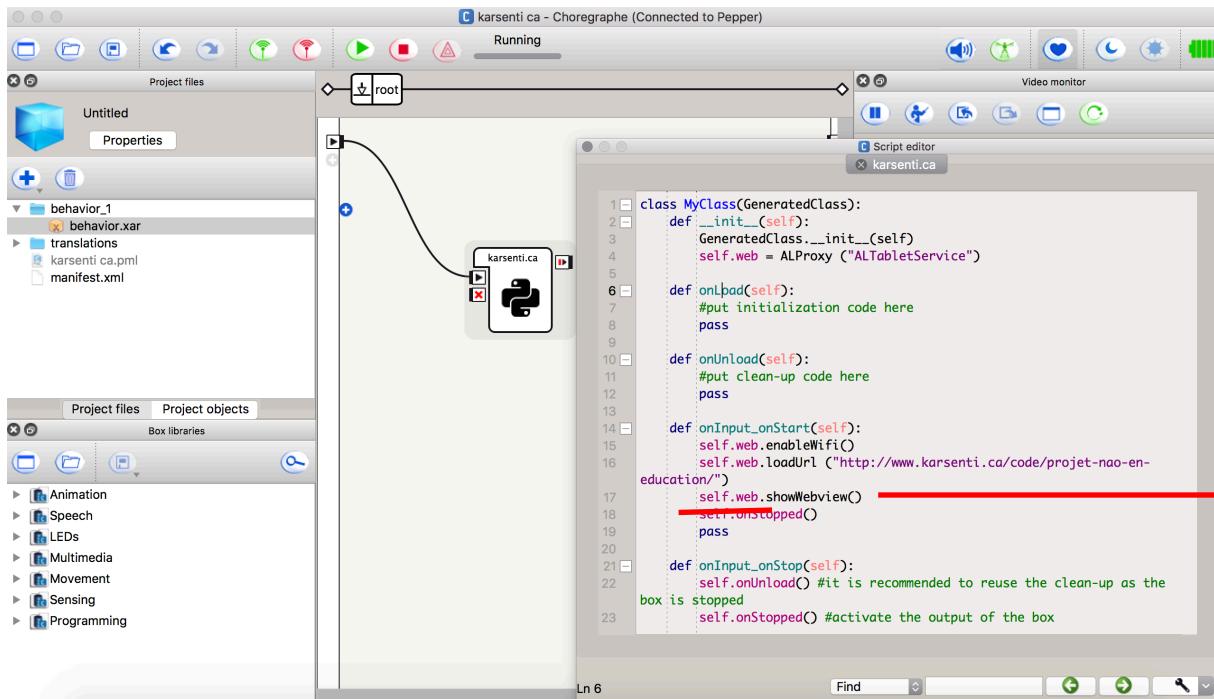
    def onLoad(self):
        #put initialization code here
        pass

    def onUnload(self):
        #put clean-up code here
        pass

    def onInput_onStart(self):
        self.web.enableWifi()
        self.web.loadUrl ("http://www.karsenti.ca/code/projet-nao-en-education/")
        self.web.showWebview()
        self.onStopped()
        pass

    def onInput_onStop(self):
        self.onUnload() #it is recommended to reuse the clean-up as the box is stopped
        self.onStopped() #activate the output of the box
```

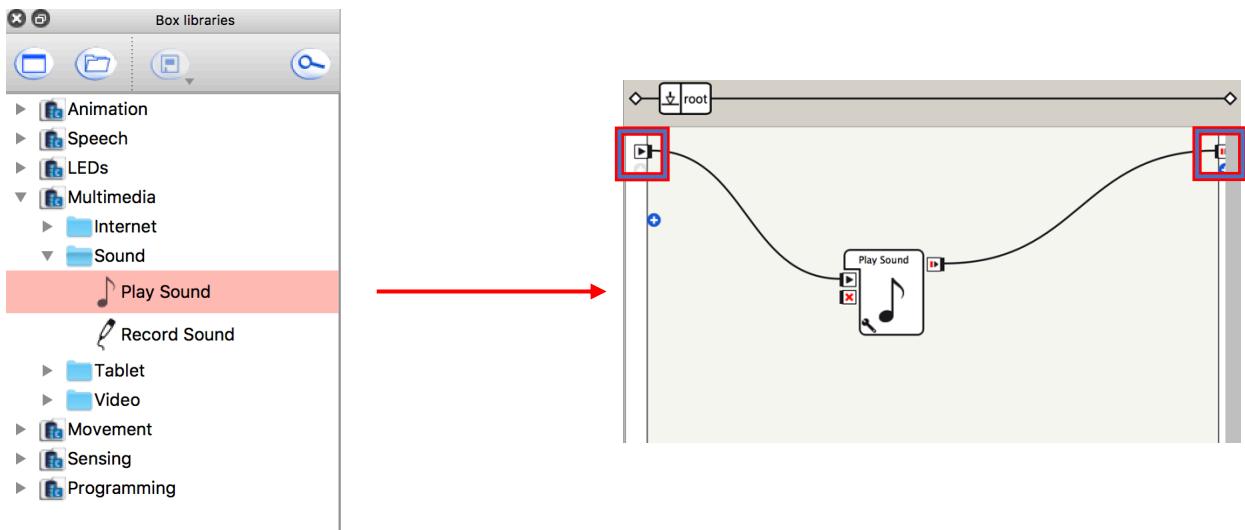
3. Tapez l'adresse internet que vous souhaitez entre les parenthèses après `self.web.loadUrl`



Attention votre Pepper doit être connecté à internet.

### 3.4 Jouer de la musique

1. Dans la fenêtre *Box libraries*, cherchez le dossier MULTIMEDIA, puis SOUND.



2. Sélectionnez *Play Sound* et déplacez la boîte dans votre espace de travail.

- N'oubliez pas de connecter ta boîte à *onStart* et à *onStop*.



3. Appuyez sur pour modifier les paramètres :



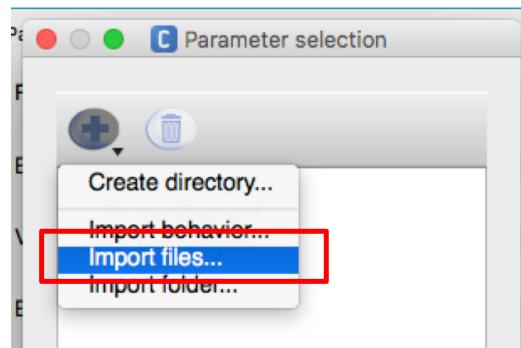
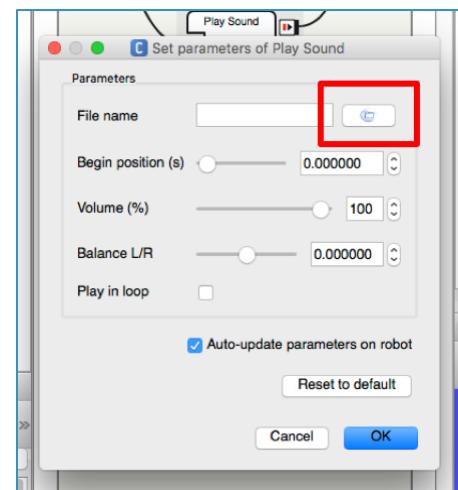
Une fenêtre va apparaître :



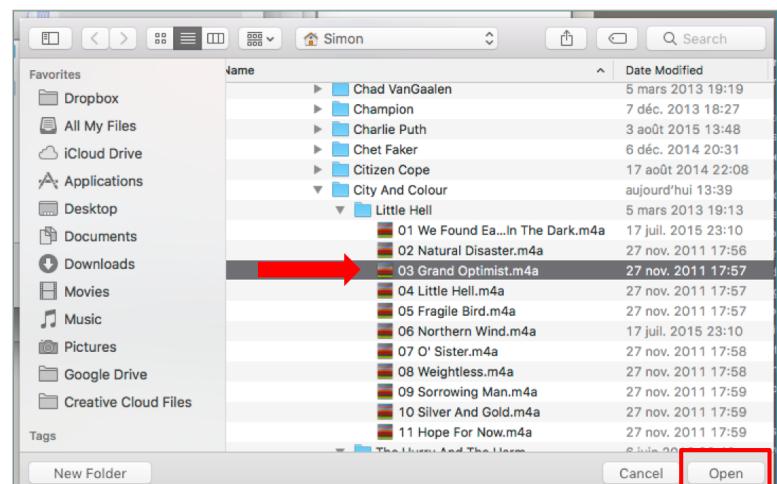
4. Appuyez sur pour chercher un fichier (une musique).



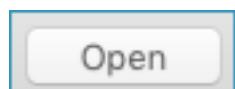
5. Appuyez sur et choisis l'option *Import files...*



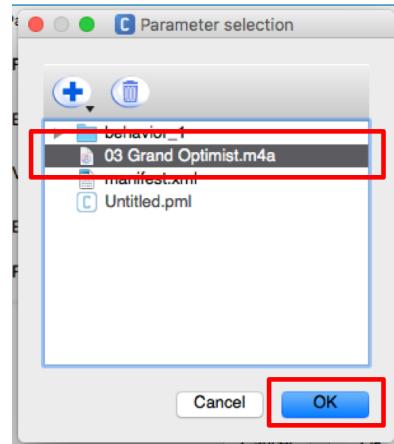
6. Vous pouvez maintenant choisir un fichier dans votre ordinateur.



7. Sélectionnez le fichier et appuyez sur

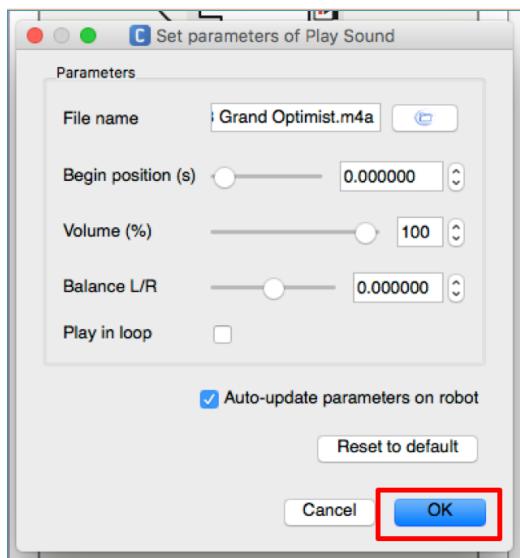


Votre fichier apparaît maintenant dans votre projet :



8. Assurez-vous qu'il soit sélectionné et appuyez sur *OK*.

Votre fichier est maintenant ajouté.



9. Appuyez sur *OK*.

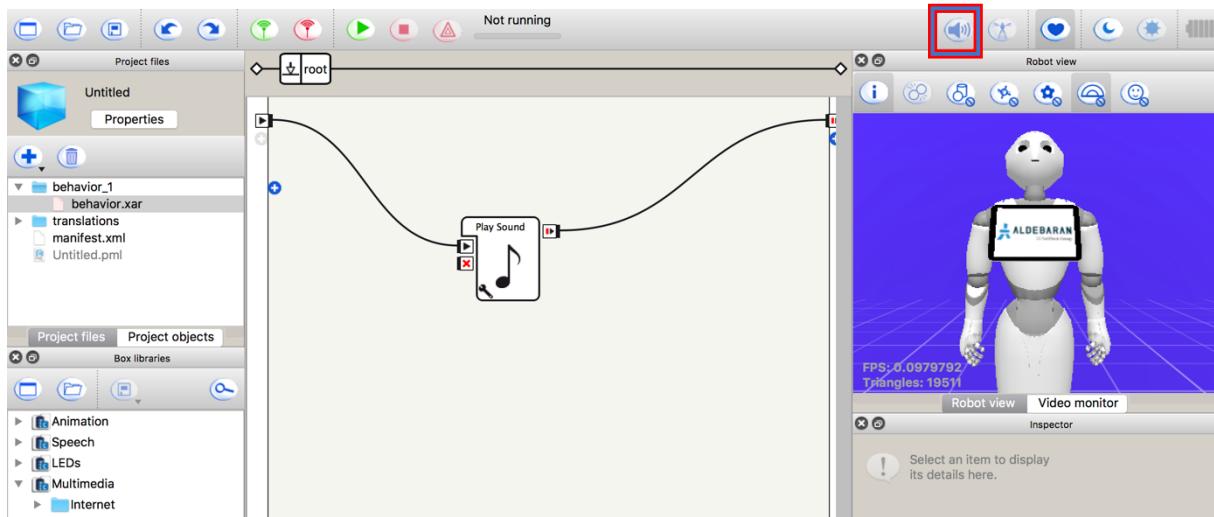


10. Pour tester ta séquence, appuyez sur

Vous devez tester la séquence avec le vrai robot pour entendre la voix.



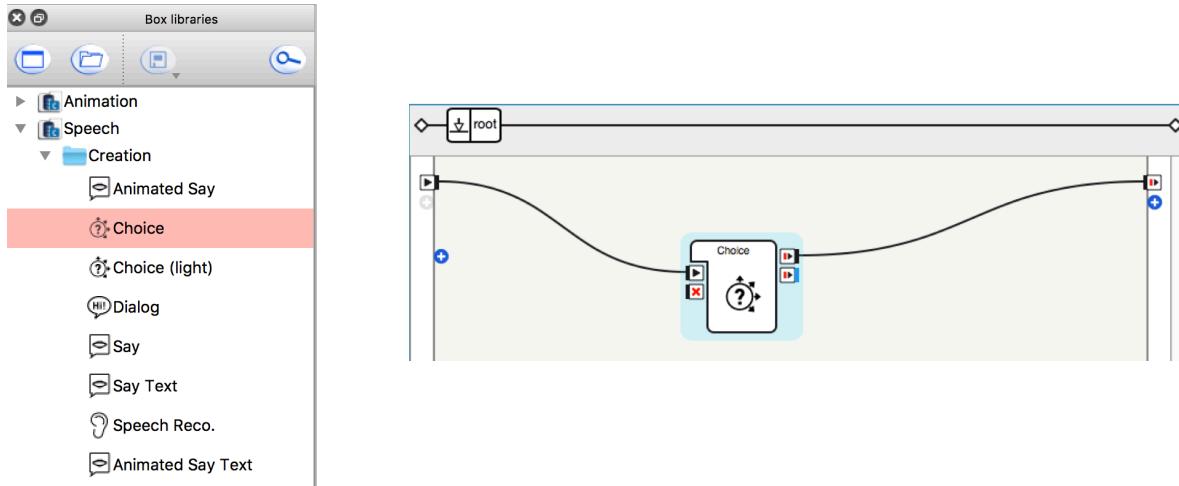
Vous pouvez régler le volume en cliquant sur le bouton



### 3.5 Poser des questions et valider les réponses

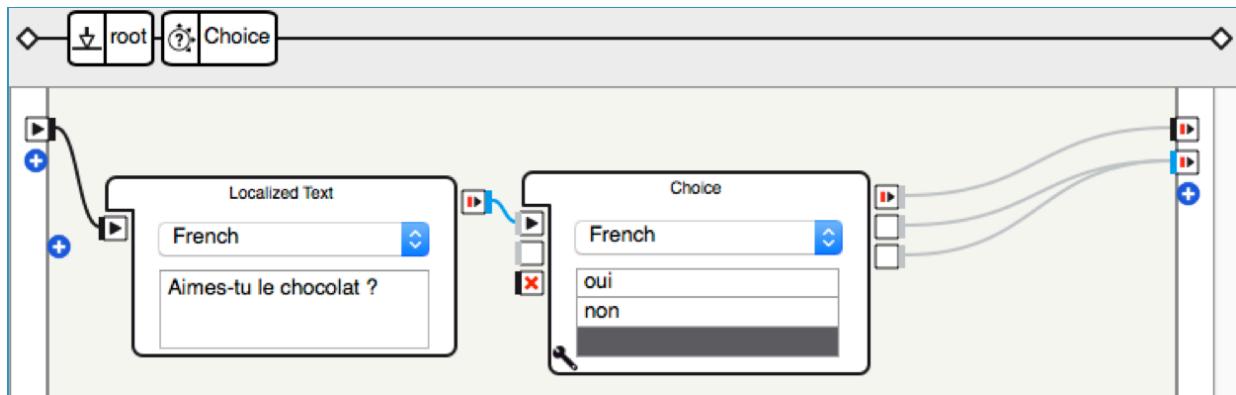
Vous pouvez utiliser la reconnaissance vocale de PEPPER pour qu'il comprenne ce que l'humain répond à ses questions et qu'il commente les réponses.

1. Pour faire poser une question à PEPPER, déposez la boîte *Choice* dans votre espace de travail :



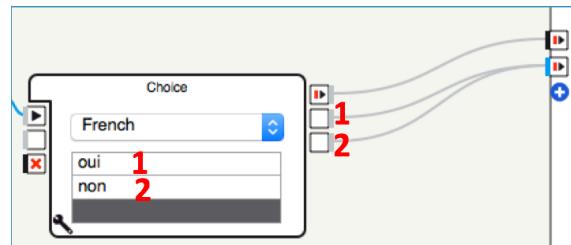
2. Double-cliquez sur la boîte *Choice*.

À l'intérieur, il y a deux boîtes : *Localized Text* et *Choice*.



3. Dans la boîte *Localized Text*, écrivez la question que PEPPER doit poser. N'oubliez pas de choisir l'option *French* pour indiquer que votre question est en français.

4. Dans la boîte *Choice*, écrivez les réponses que PEPPER va pouvoir reconnaître.



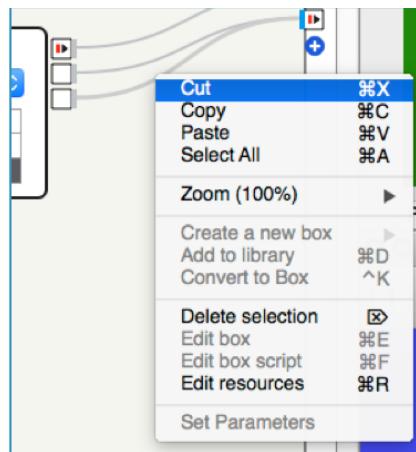
Comme vous pouvez le voir, il y a deux lignes (2 réponses) et deux outputs différents. Donc, si PEPPER entend « oui », il va envoyer un signal à l'output 1.

Pour faire une réponse personnalisée, vous devez supprimer le lien entre l'output et le *onStopped output* (le petit carré avec la ligne bleue) :

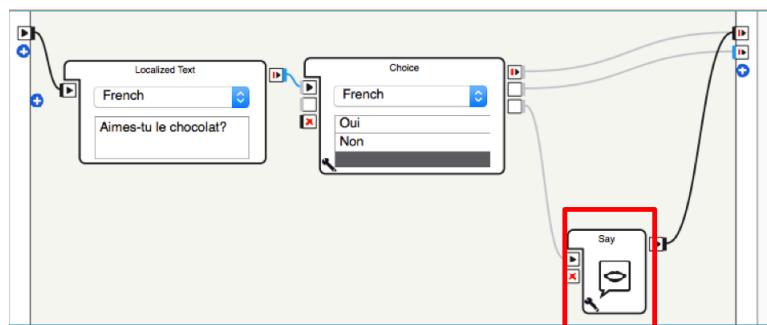
5. Pour supprimer le lien, sélectionnez-le (il va devenir bleu) :



6. Vous pouvez ensuite faire un clique-droit avec votre souris et sélectionner l'option *Cut* :

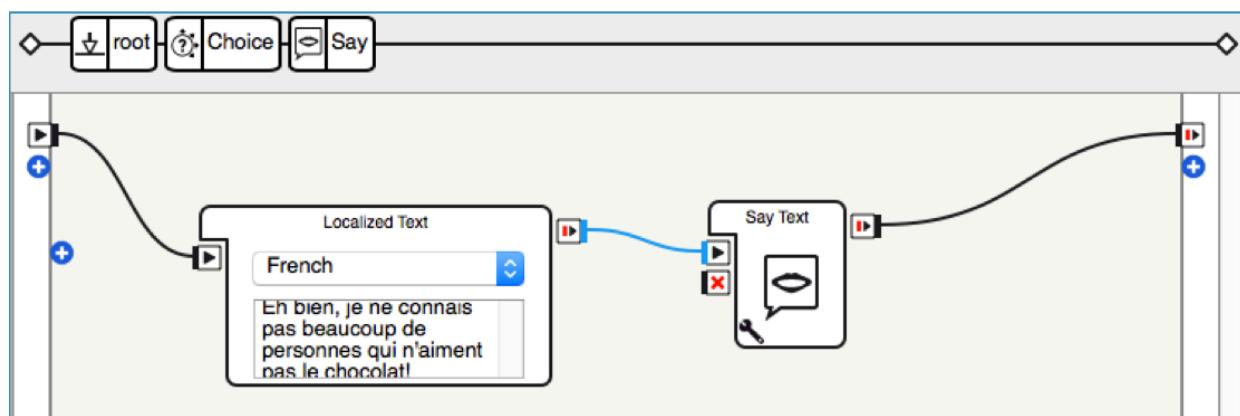


7. Maintenant que le lien est supprimé, vous pouvez ajouter une boîte *Say* et la relier comme ci-dessous :



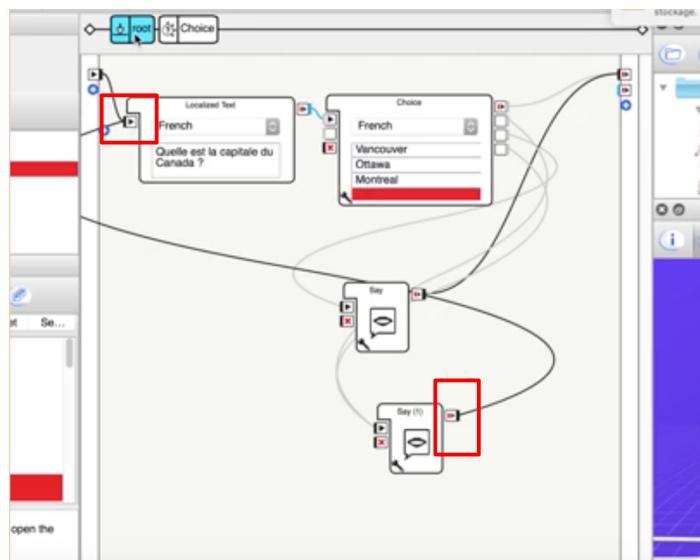
\* IMPORTANT : remarquez bien que l'output de la boîte *Say* n'est plus connecté à l'*output onStopped* avec la ligne bleue, mais bien à celui avec la ligne noire, juste au-dessus.

8. Double-cliquez sur la boîte *Say* pour indiquer ce que PEPPER doit dire si on répond « Non ».



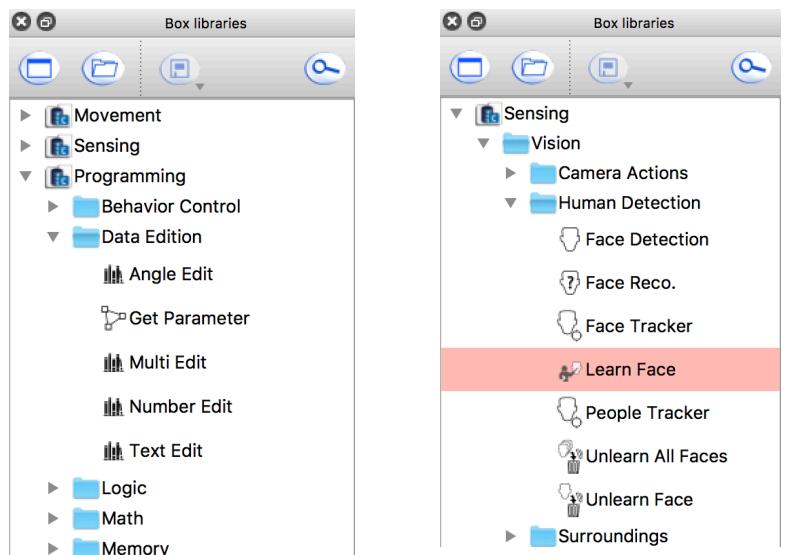
9. Vous pouvez faire la même opération pour la réponse « Oui ».

10. Quand vous voulez que PEPPER vous repose la question parce que vous avez donné une réponse incorrecte, vous devez connecter l'output de la boîte *Say*, connectée aux différentes mauvaises réponses, à l'input de la boîte *Localized text*.

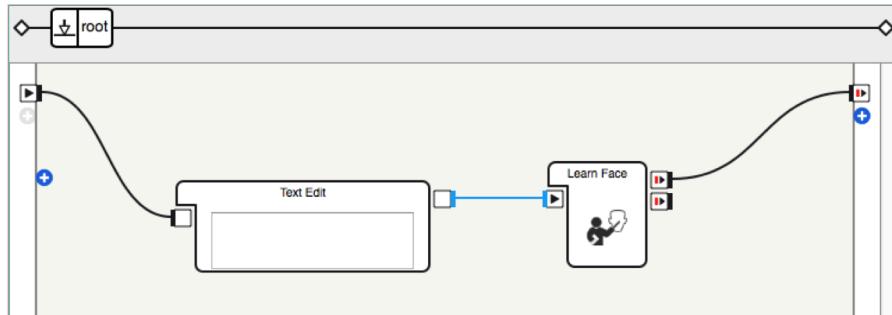


### 3.6 Reconnaître des personnes

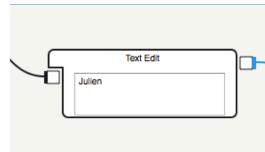
1. Vous devez d'abord sélectionner les boîtes *Text Edit* et *Learn Face*, puis les déposer dans votre espace de travail en les reliant à *onStart* et à *onStop* :



Votre espace de travail doit ressembler à ceci :

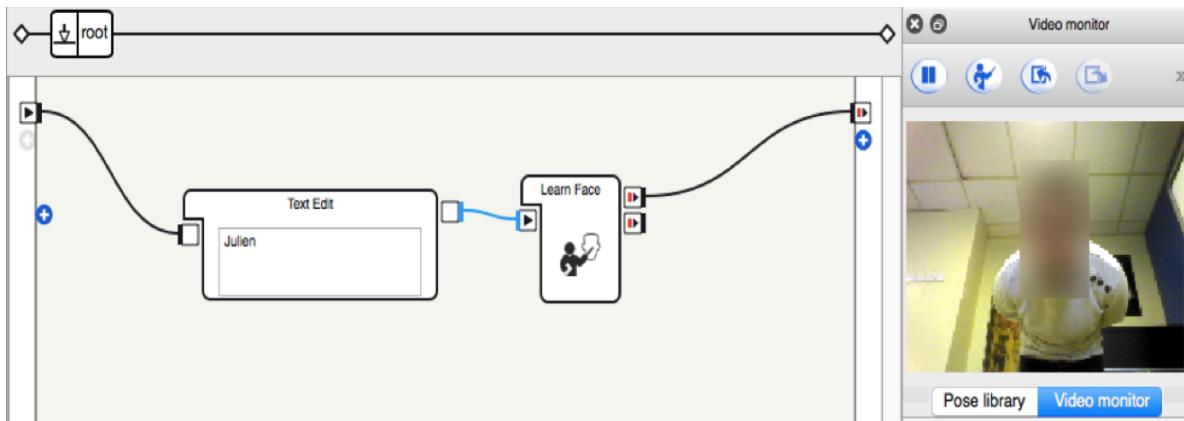


2. Écrivez le nom de la personne dont vous souhaitez enregistrer le visage dans la boîte *Text Edit*.



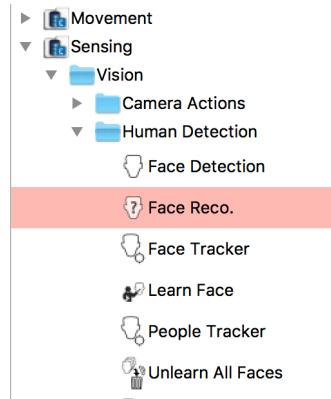
3. Ouvrez la fenêtre *Video monitor*.

4. Vous pouvez ensuite lancer la séquence (bouton *Play*) et demander à la personne dont vous souhaitez enregistrer le visage de se placer devant les yeux de PEPPER



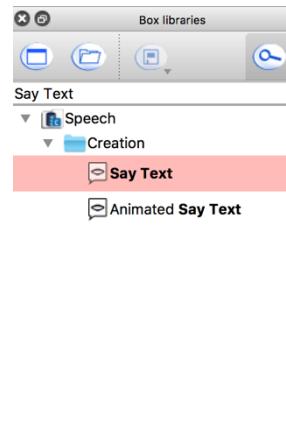
Si PEPPER a bien enregistré le visage, ses yeux deviendront verts. Si ses yeux deviennent rouges, c'est que quelque chose n'a pas fonctionné. Il faut recommencer la manipulation.

5. Maintenant que le visage est enregistré, vous pouvez le faire reconnaître par PEPPER. Pour ce faire, tu dois utiliser la boîte *Face Reco.*

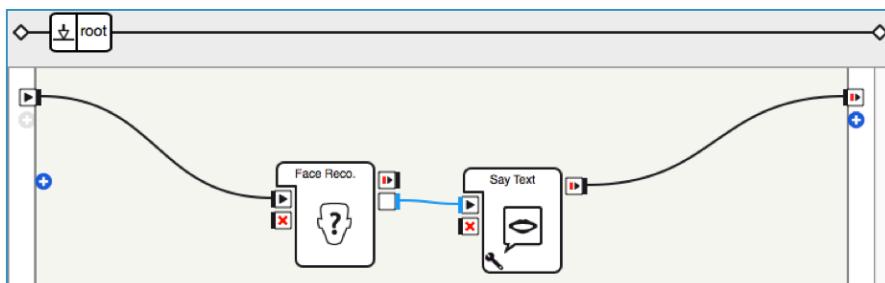


6. Pour que PEPPER dise le nom de la personne quand il la voit, vous devez aussi mettre la boîte *Say Text*.

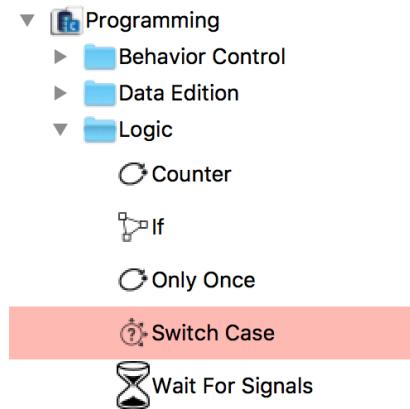
7. Cette boîte est dans le dossier SPEECH et le sous-dossier CREATION.



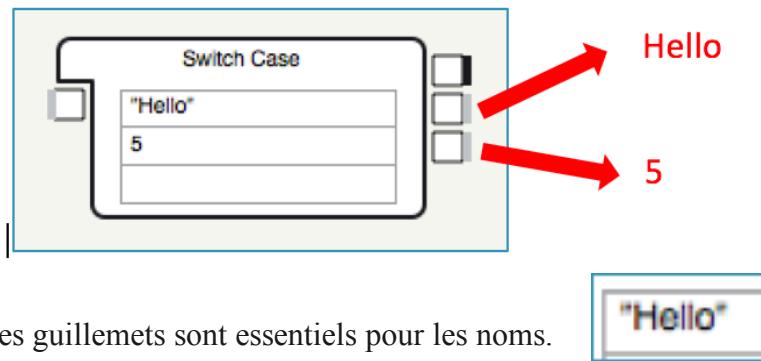
Votre espace de travail devrait ressembler à ceci :



8. Si vous voulez lui faire reconnaître plusieurs visages et faire dire un message personnalisé à PEPPER, vous pouvez utiliser la boîte *Switch Case* :

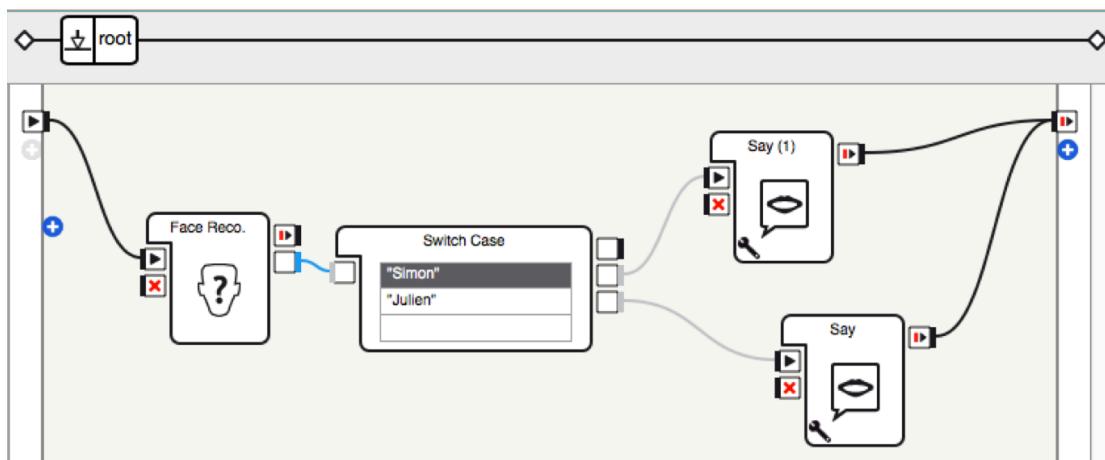


9. Comme vous pouvez voir, pour chaque ligne (qui correspond à une information), il y a un output :

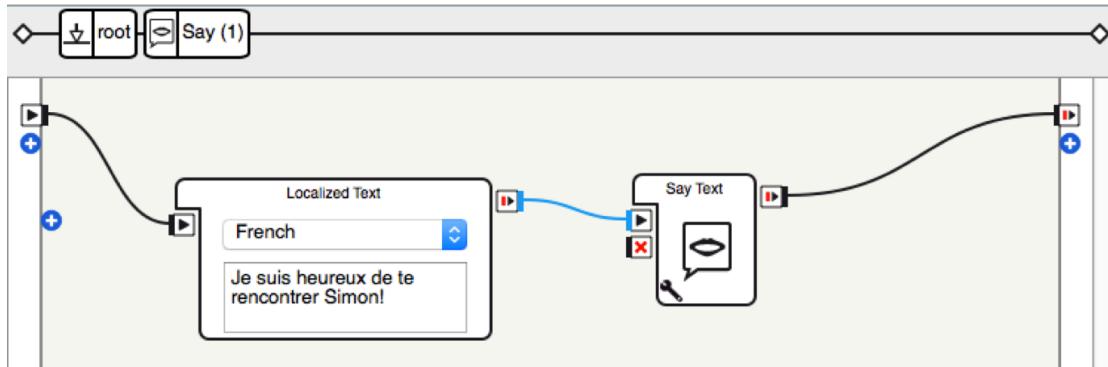


\* IMPORTANT : les guillemets sont essentiels pour les noms.

10. Vous pouvez donc écrire les noms des personnes pour qui vous avez enregistré les visages et ajouter des boîtes *Say* pour mettre des messages personnalisés :



11. Ensuite double-cliquez sur les boîtes *Say* pour entrer le message que vous voulez faire dire à PEPPER :



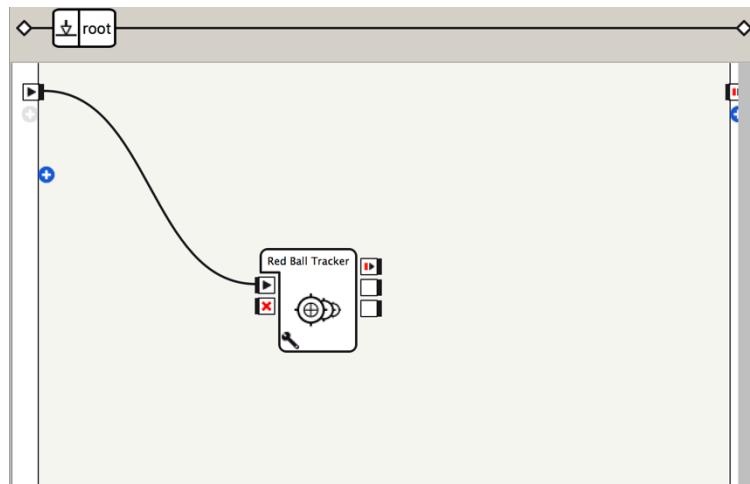
### 3.7. Reconnaître des objets

#### 3.7.1 Suivre la balle rouge

1. PEPPER sait déjà reconnaître la balle rouge : vous n'avez donc pas besoin de lui faire apprendre cet objet. Il faut utiliser la boîte *Red Ball Tracker* située dans le dossier SENSING, le sous-dossier VISION et le sous-dossier SURROUNDINGS.

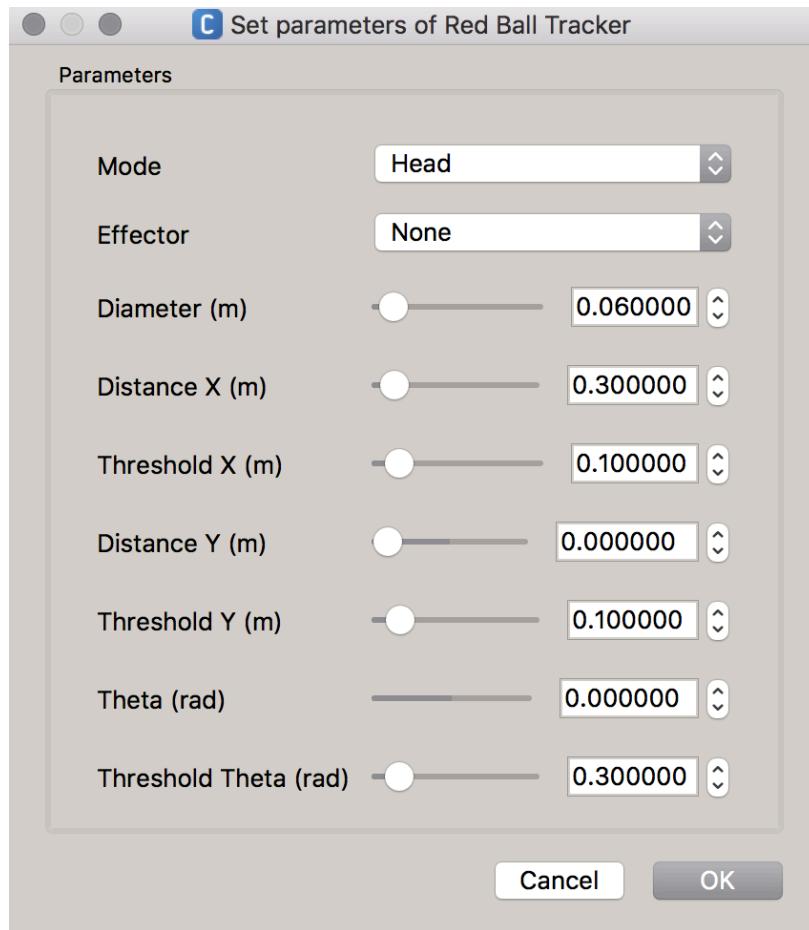
▼ Sensing  
  ▼ Vision  
    ▼ Surroundings  
       Red Ball Tracker

2. Vous pouvez la déposer dans votre espace de travail et la lier à l'Input *onStart*.

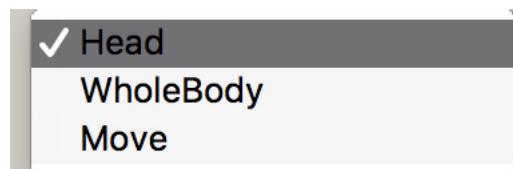


3. Il y a plusieurs options pour cette boîte. Vous pouvez accéder aux options en appuyant sur la clé à molette dans le coin en bas à gauche de la boîte.

Une nouvelle fenêtre va apparaître :



4. L'option *Mode* vous permet de choisir parmi 3 fonctions :



- *Head* : PEPPER bouge seulement sa tête pour suivre la balle des yeux.
- *WholeBody* : PEPPER bouge tout son corps pour se mettre face à la balle.
- *Move* : PEPPER se déplace pour aller devant la balle rouge.

5. L'option *Effector* vous permet de choisir parmi 4 fonctions :



- *None* : Aucune action supplémentaire.

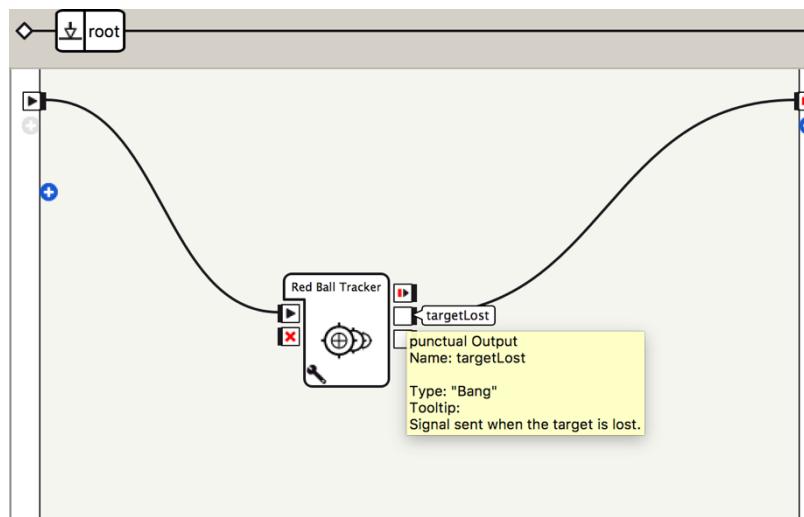
- *Arms* : PEPPER lève ses deux bras.
- *LArm* : PEPPER lève son bras gauche.
- *RArm* : PEPPER lève son bras droit.

6. Le paramètre Distance X (m) vous permet de déterminer à quelle distance de la balle PEPPER doit s'arrêter

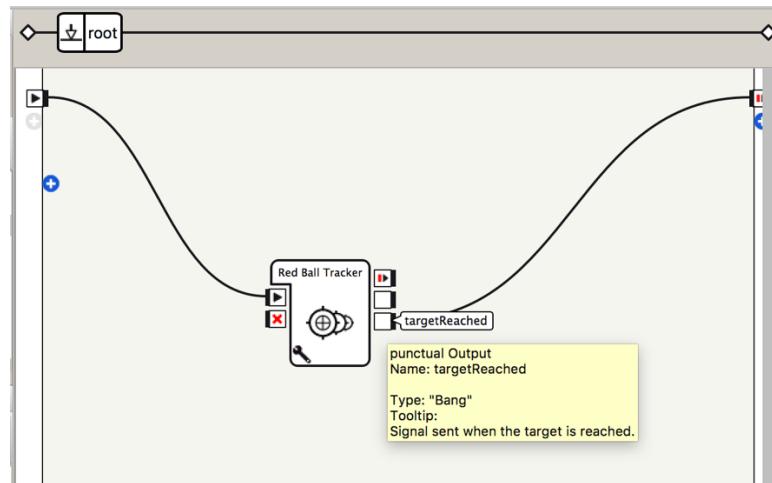


7. Deux possibilités s'offrent à vous pour mettre fin au comportement :

- Vous pouvez relier l'output *targetLost* à l'output *OnStopped* si vous souhaitez arrêter le comportement quand la balle n'est plus dans le champ de vision du robot.



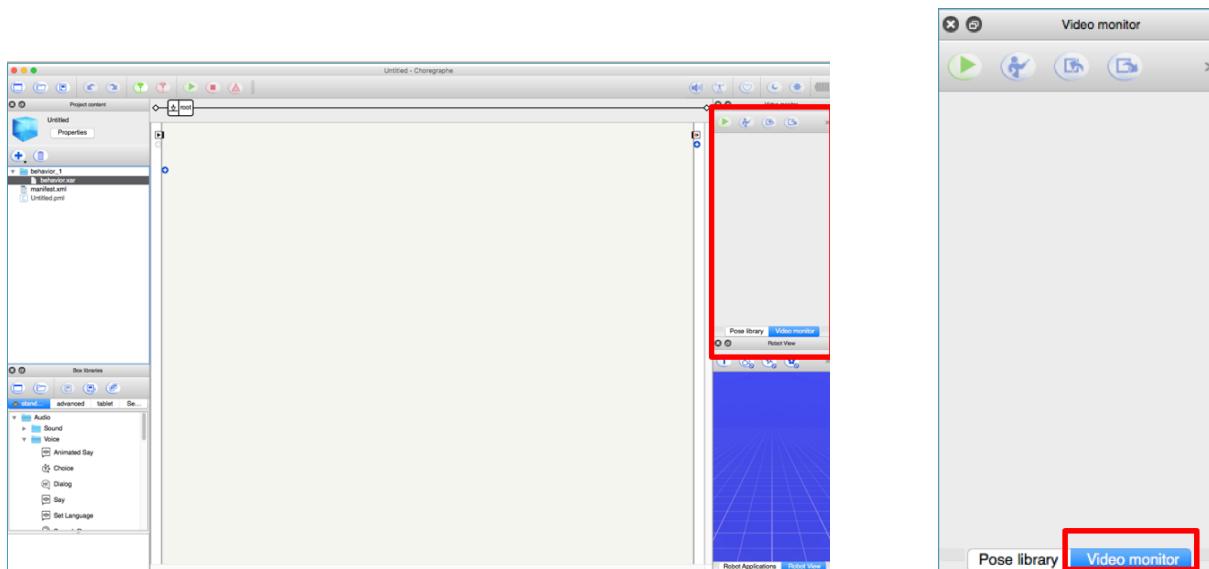
- Vous pouvez relier l'output *targetReached* à l'output *OnStopped* si vous souhaitez que le comportement s'arrête quand le robot atteint son objectif (par exemple : s'arrêter à 20 centimètres de la balle).



### 3.7.2 Apprendre à reconnaître des objets

Vous pouvez apprendre à PEPPER à reconnaître les choses devant lui. En utilisant la fenêtre *Video monitor*, votre robot peut apprendre des images, des objets ou des lieux (salle de classe).

1. D'abord, assurez-vous que vous êtes connecté à un vrai robot ou à un robot simulé dans un univers virtuel.



La fenêtre va maintenant afficher ce que la caméra voit à ce moment-là.

2. Appuez sur le bouton Play  puis sur le bouton *Learn* .

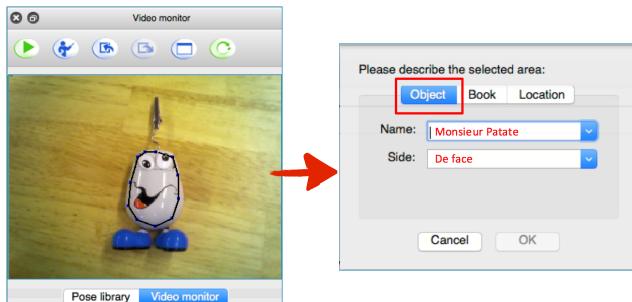
Un décompte de 4 secondes va commencer : il vous permet de placer correctement l'objet.

À zéro, l'image est capturée.



3. Cliquez pour dessiner, segment par segment, le contour de l'objet que vous voulez faire apprendre à PEPPER.

Au moment où vous cliquerez à nouveau sur le premier point pour fermer votre contour, une fenêtre s'ouvrira :



4. Nommez l'objet que vous avez photographié. Si c'est un objet, restez dans l'onglet *Object*. Si c'est un livre, allez dans l'onglet *Book* et si c'est un lieu cliquez sur *Location*.

5. Quand vous avez terminé d'entrer les informations, appuez sur *OK*.

Un message va apparaître pour confirmer si l'apprentissage a fonctionné ou pas. Il est possible que l'apprentissage ne fonctionne pas si l'image n'est pas de bonne qualité, si elle est floue, etc.

Il est conseillé que votre objet soit sur un fond uni de couleur différente. Par exemple, pour apprendre à PEPPER à reconnaître un objet bleu, un fond uni blanc conviendrait très bien.

6. Si vous voulez faire apprendre d'autres objets à PEPPER, appuie sur  et recommencez les mêmes étapes.

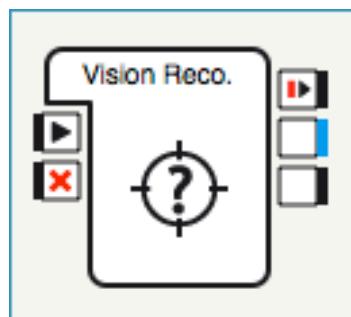
Quand vous avez terminé, vous pouvez :

- Appuyer sur  (*Send current vision recognition database to the robot*) pour utiliser immédiatement cette base de données avec le robot ;
- Appuyer sur  (*Export Vision Recognition Database*) pour enregistrer la base de données sur ton ordinateur pour l'utiliser plus tard.

Maintenant, PEPPER est capable de reconnaître cet objet quand il le voit.

Pour chercher une base de données (*database*) créée auparavant, appuez sur .

7. Vous devez créer un comportement en utilisant la boîte *Vision Reco* :



8. Débutez ensuite la séquence pour que PEPPER puisse reconnaître les objets devant lui.

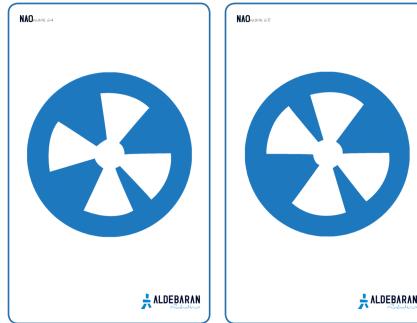


Cette méthode est difficile à réaliser. PEPPER a beaucoup de difficultés à reconnaître les objets. Des conditions optimales sont nécessaires, c'est-à-dire une très bonne luminosité et un fond uni blanc. Les NAOMarks sont une bonne alternative.

### 3.7.3 Utiliser les NAOMarks

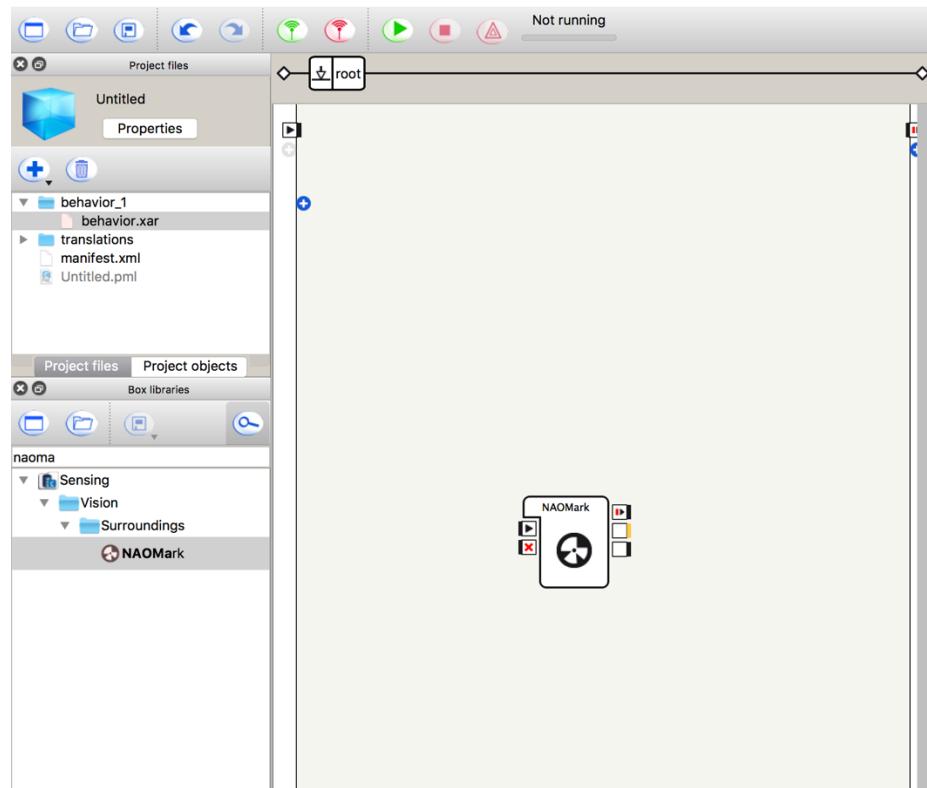
Vous allez découvrir l'utilisation des NAOMarks.

Les NAOMarks sont des repères visuels enregistrés dans PEPPER. Ils ressemblent à des ventilateurs. Ce sont des cercles noirs ou bleus avec à l'intérieur un ensemble de triangles dirigés vers le centre. Chaque NAOMark est spécifique et porte un numéro.



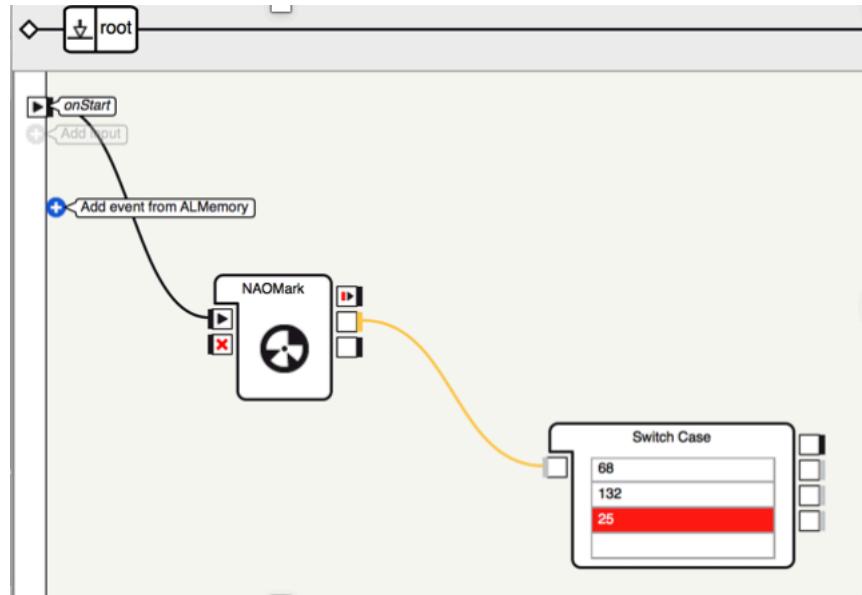
Vous pouvez placer ces repères à différents endroits dans le champ de vision de PEPPER. Vous pouvez même les coller sur des objets ou à des localisations précises.

1. Ajoutez la boîte NAOMark dans votre espace de travail, que vous trouverez dans le dossier SENSING, sous-dossier VISION et SURROUNDINGS.

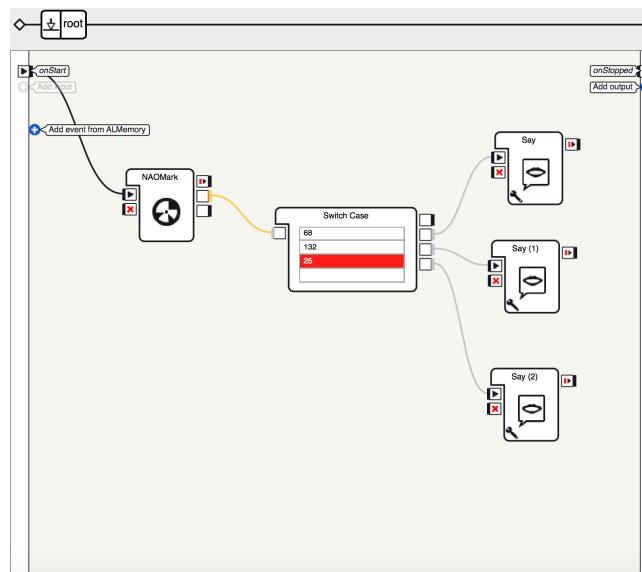


Quand NAO détecte un NAOMark, son numéro s'affiche au niveau de la sortie jaune.

3. Après, glissez la boîte *Switch case*, connecte-la à la sortie jaune et tape les numéros des NAOMarks.



4. Vous pouvez maintenant ajouter un comportement pour chaque NAOMark.



5. Pour terminer, testez votre séquence.

## 4. Exemples de créations d'applications

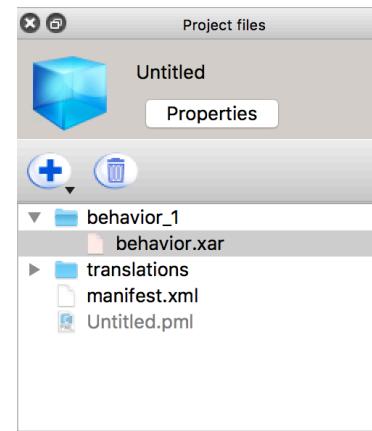
### 4.1. L'histoire du Petit chaperon rouge

PEPPER est capable de raconter des histoires de manière animée et en affichant les images.

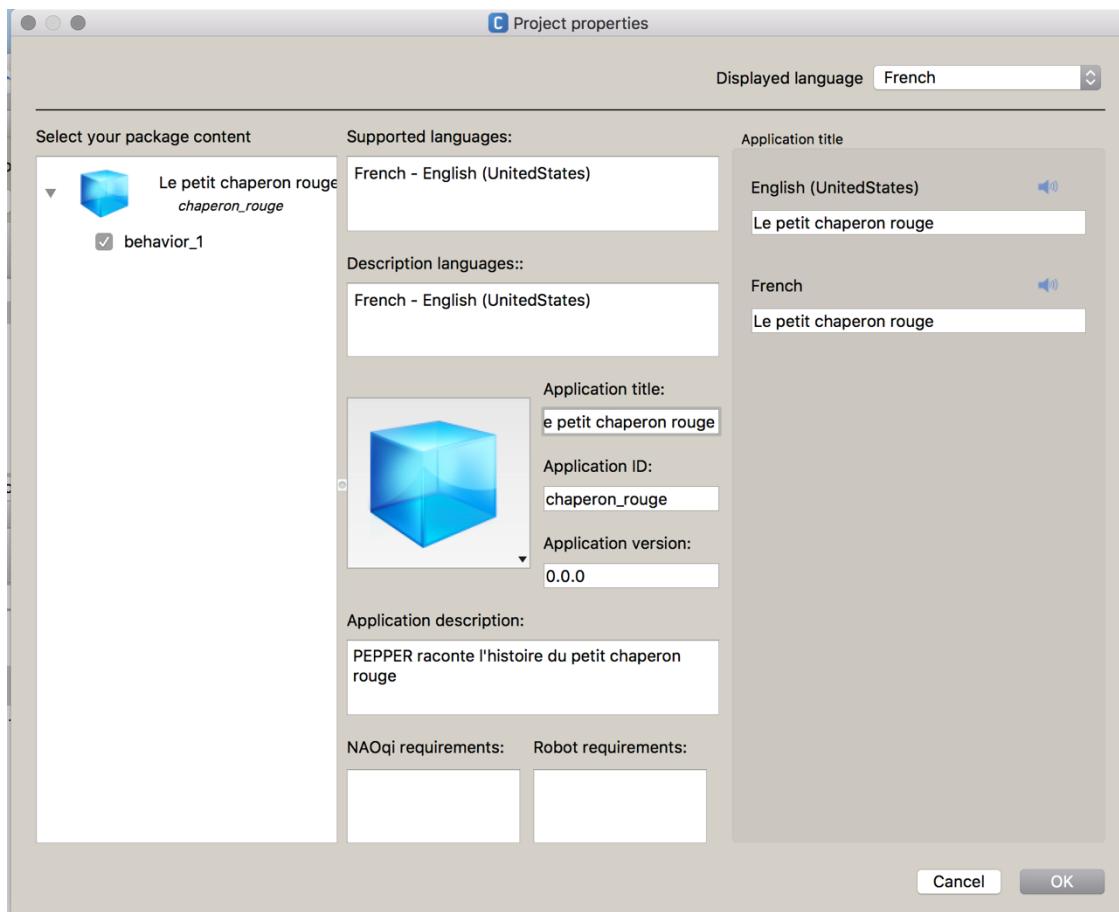
1. Ouvrez Choregraphe

2. Cliquez sur *Properties* dans la fenêtre *Project files*

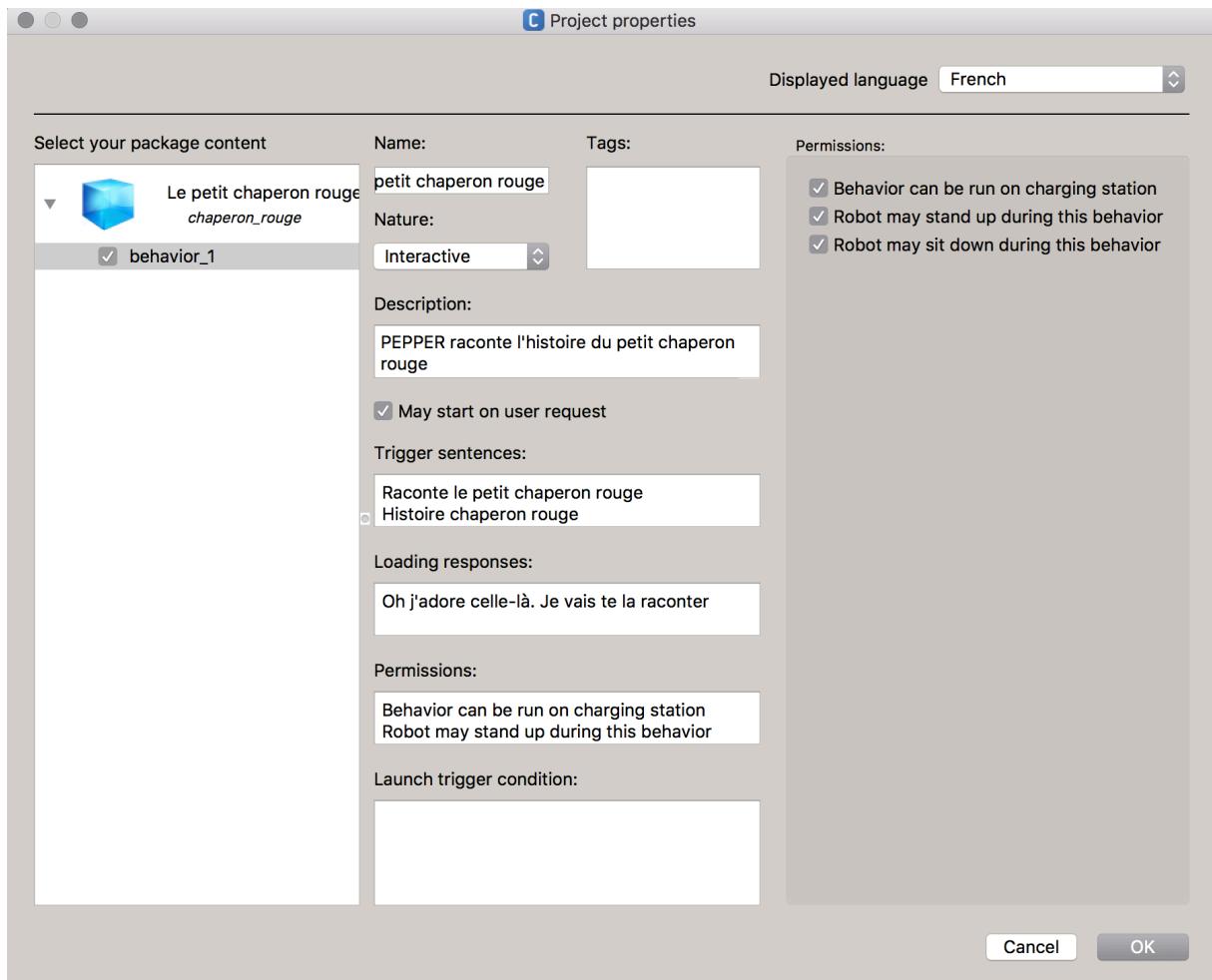
3. Définissez les propriétés de votre projet : Choisissez la langue, mettez un titre, une description et une image. L'image apparaîtra sur la tablette de PEPPER.



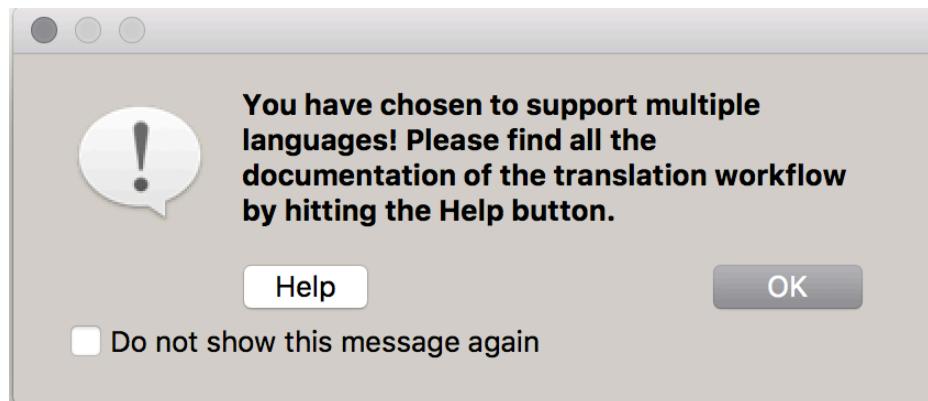
Vous êtes obligé de recopier votre titre français dans l'onglet *English (UnitedStates)* pour le voir apparaître comme titre du fichier sur l'interface de Choregraphe et dans la tête du robot.



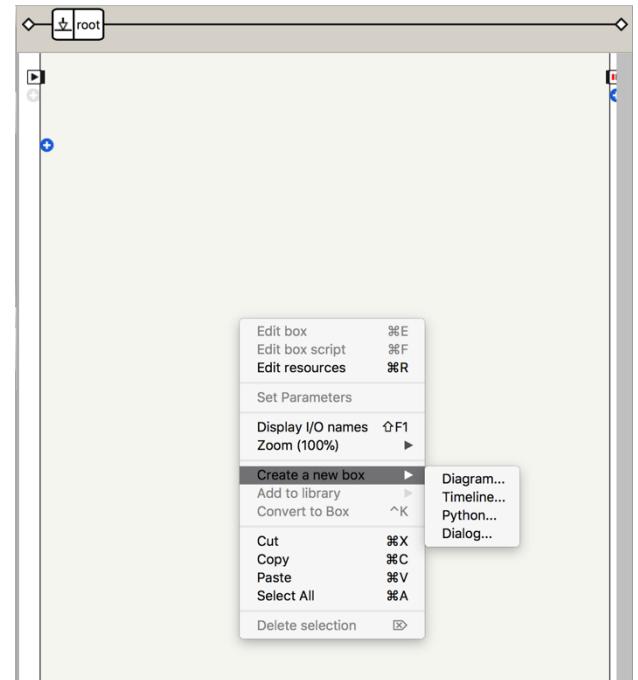
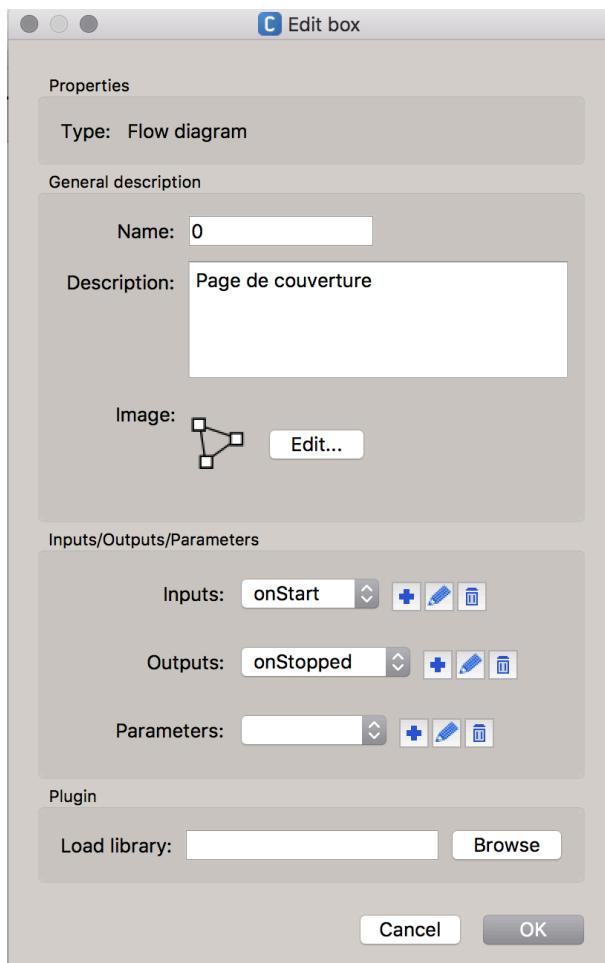
4. Cliquez sur *behavior\_1* les paramètres du comportement : Nommez le, choisissez sa nature, créez des phrases de déclenchement, des réponses et donnez des permissions.



5. Cliquez sur Ok

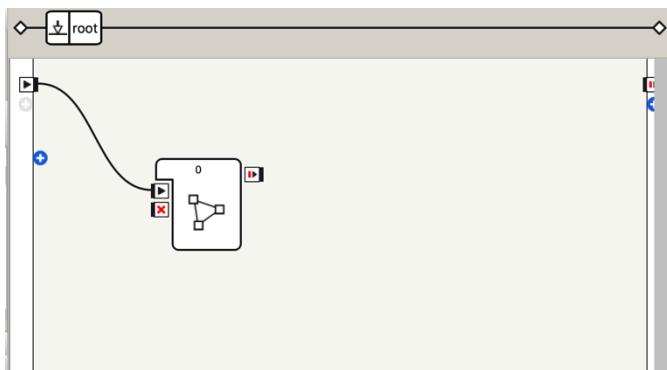


6. Créez une boîte *Diagram*. Cliquez droit dans votre espace de travail. Allez sur *Create a new box* et appuyez sur *Diagram...* Vous aurez besoin d'autant de boîtes *Diagram* que de pages dans votre livre.

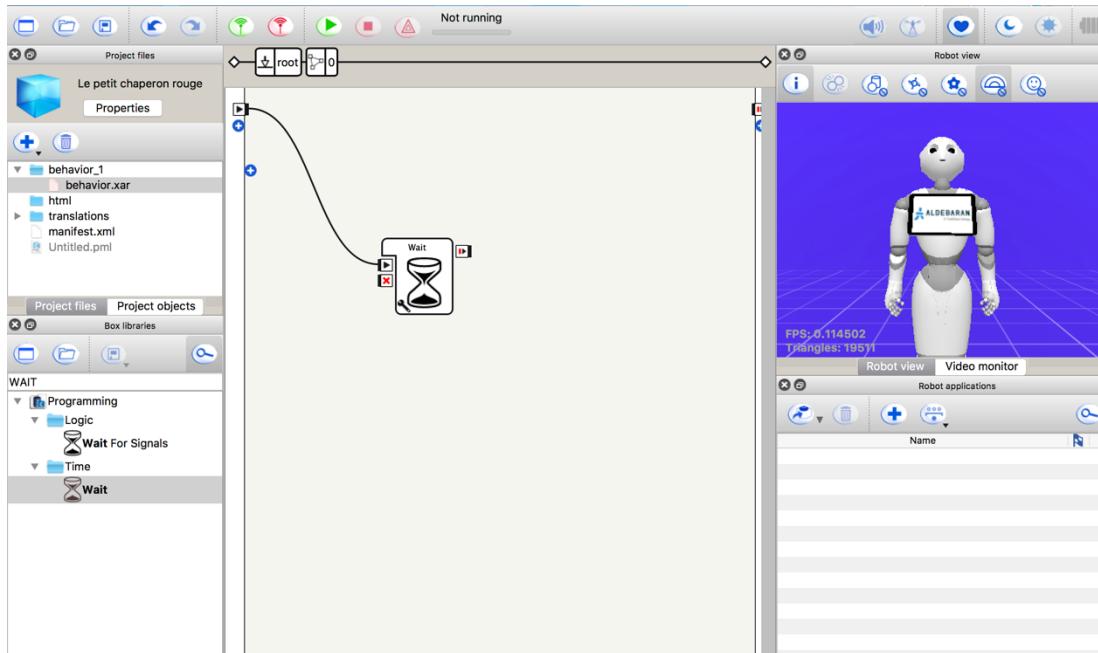


7. La fenêtre *Edit box* apparaît. Donnez un nom et une description à votre boîte, puis cliquez sur *Ok*. Dans cet exemple, la boîte *0* correspond à la page de couverture, c'est-à-dire que lorsque PEPPER va jouer cette boîte, il va introduire le livre en parlant, en bougeant et en montrant l'image de la page de couverture du Petit chaperon rouge.

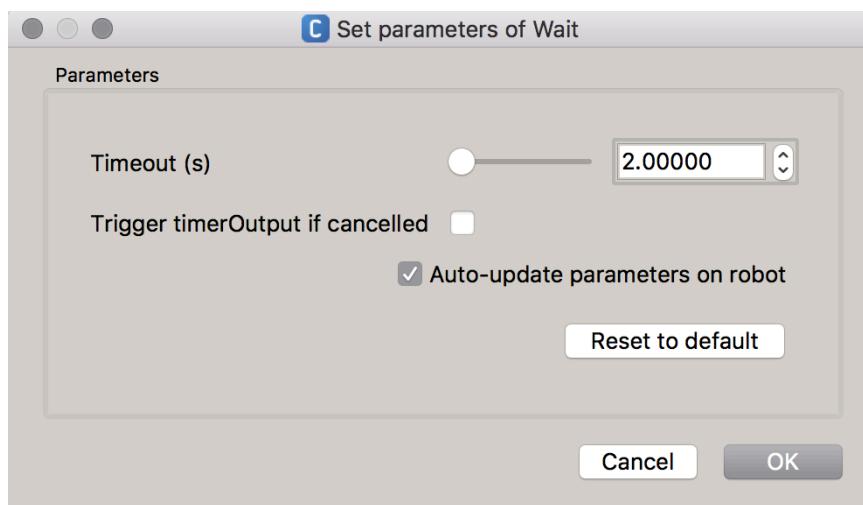
8. La boîte est maintenant dans votre espace de travail. Double-cliquez dessus pour y ajouter les comportements. Vous apercevez sur la ligne de temps que vous êtes à l'intérieur.



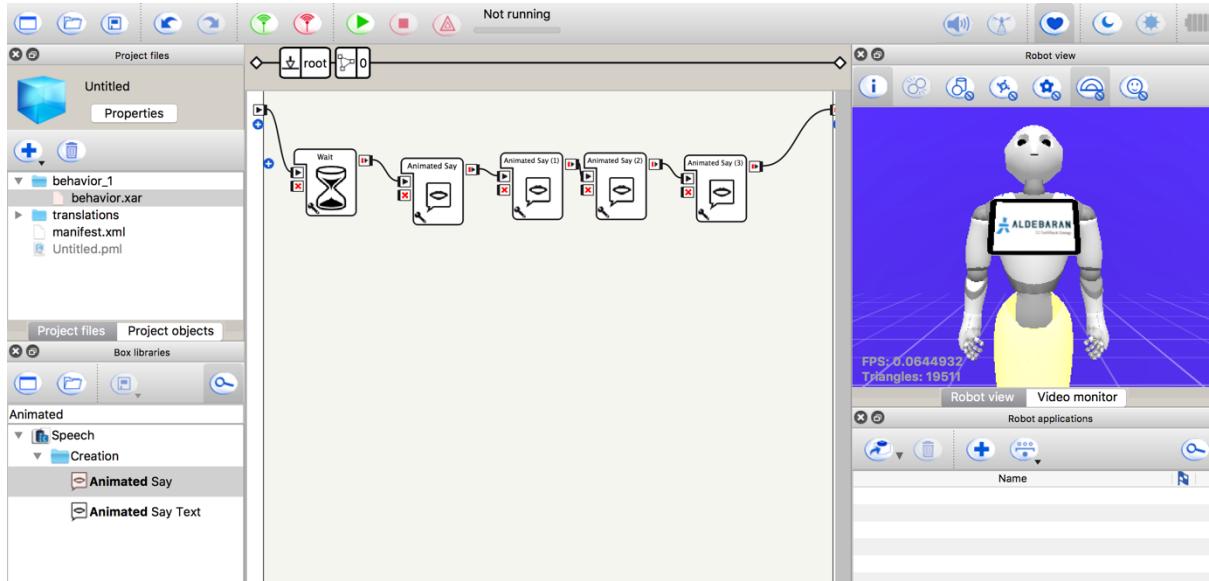
9. Ajoutez, au début de votre programme, la boîte *Wait* que vous trouverez dans le dossier PROGRAMMING, dans le sous-dossier TIME. Cela vous permettra de laisser le temps au robot d'afficher l'image sur la tablette avant d'énoncer le texte.



10. Cliquez sur la clé à molette de la boîte pour définir le temps. Deux secondes sont suffisantes pour afficher une image.



11. Ajoutez des boîtes *Animated Say* pour que PEPPER raconte l'histoire. Il est conseillé d'utiliser plusieurs boîtes *Animated Say*, dans chaque boîte, pour fluidifier votre texte et varier les voix si vous avez différents personnages. Des boîtes *Wait* peuvent également être ajoutées entre les boîtes *Animated Say* pour, en plus, laisser des pauses entre certaines phrases. Je rappelle que chaque boîte créée correspond à une page du livre.



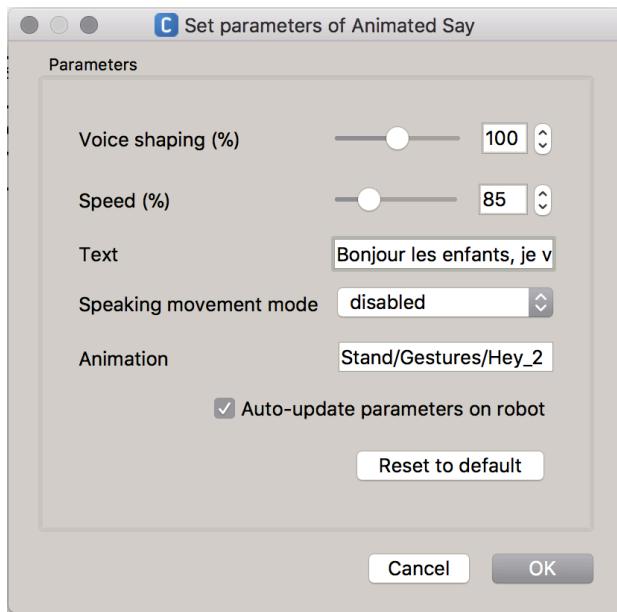
12. Cliquez sur la clé à molette de la boîte *Animated Say* afin d'ajouter le texte et régler les paramètres.

- Inscrivez ce que vous voulez que PEPPER dise dans l'onglet *Text*.
- L'onglet *Voice shaping* permet de modifier la voix du robot. Plus le nombre est faible et plus le son est grave. Inversement plus le nombre est élevé et plus le son est aigu.
- L'onglet *Speed* permet de contrôler la vitesse de parole.
- L'onglet *Speaking movement mode* permet de choisir un des trois modes de mouvement.



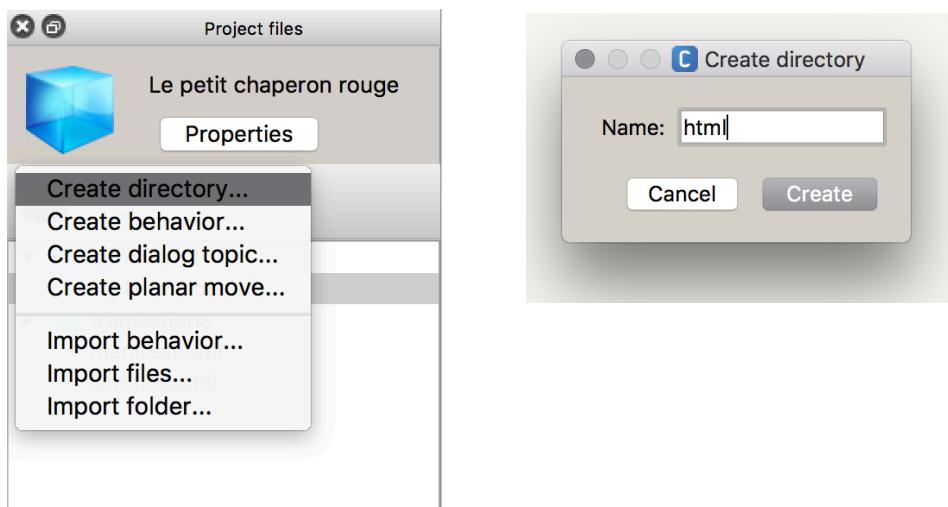
- *Disabled* permet de jouer le mouvement que tu as inséré dans l'onglet *Animation* ;
- *Random* permet de jouer des mouvements au hasard ;
- *Contextual* permet de jouer des mouvements en contexte avec ce que le robot dit.

- L'onglet *Animation* permet d'insérer le mouvement de votre choix. Vous retrouvez la bibliothèque des postures dans la bibliothèque d'Aldebaran : <http://doc.aldebaran.com/2-5/naoqi/motion/alanimationplayer-advanced.html>



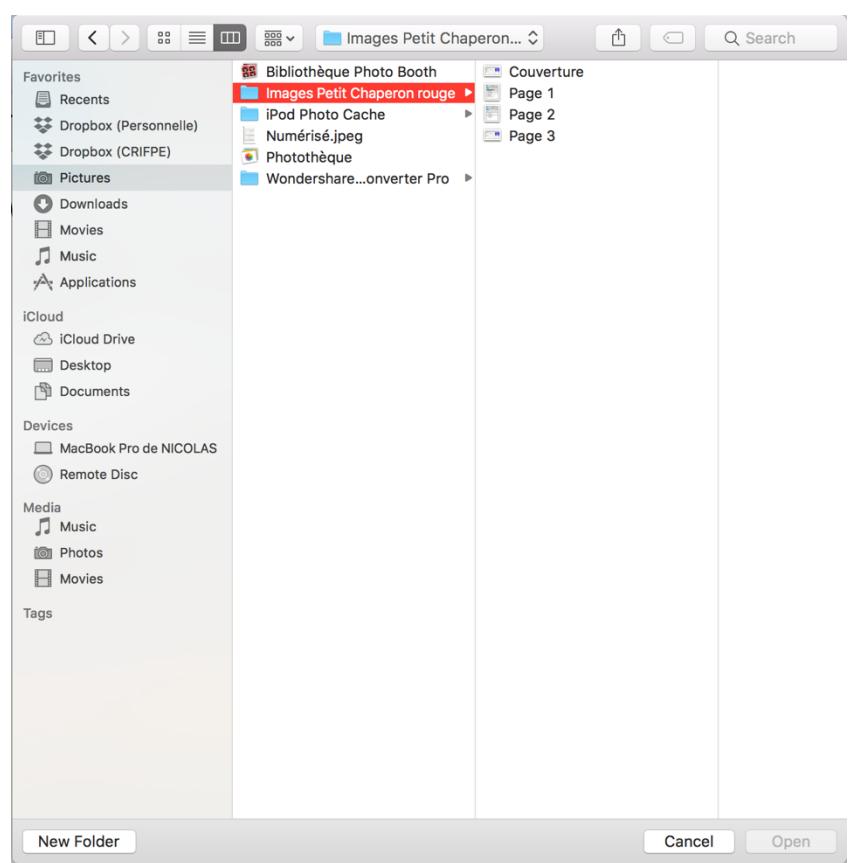
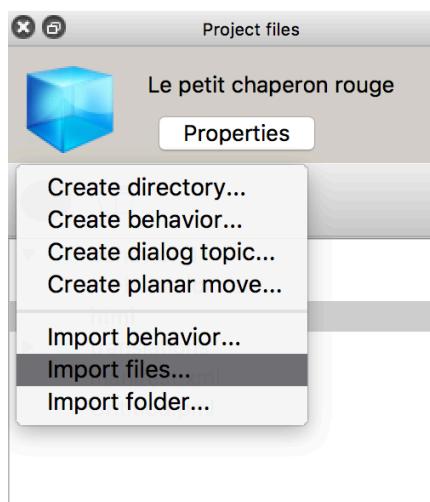
13. Maintenant vous allez ajouter l'image qui va avec le texte. D'abord, il faut créer un dossier

appelé *html*. Cliquez sur  dans la fenêtre *Project files*, puis sur *Create directory*. Nommez le dossier *html* et appuyez sur *Create*.



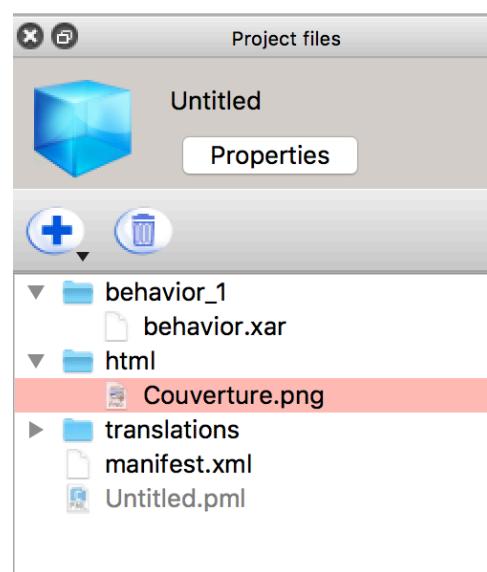
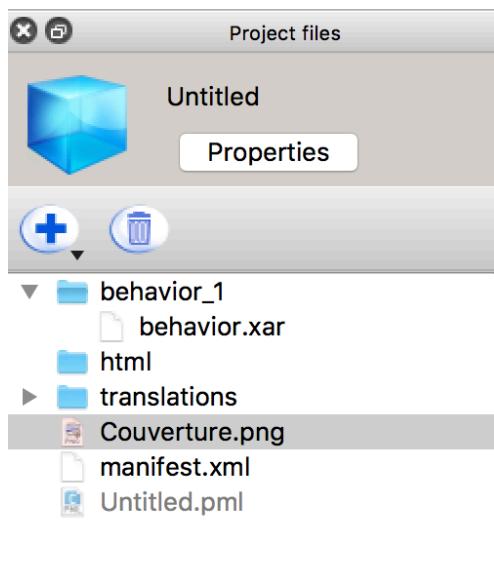
Le dossier apparaît dans la fenêtre *Project files*.

14. Vous allez ajouter l'image que vous souhaitez dans ce nouveau dossier. Cliquez sur  puis sur *Import files*.

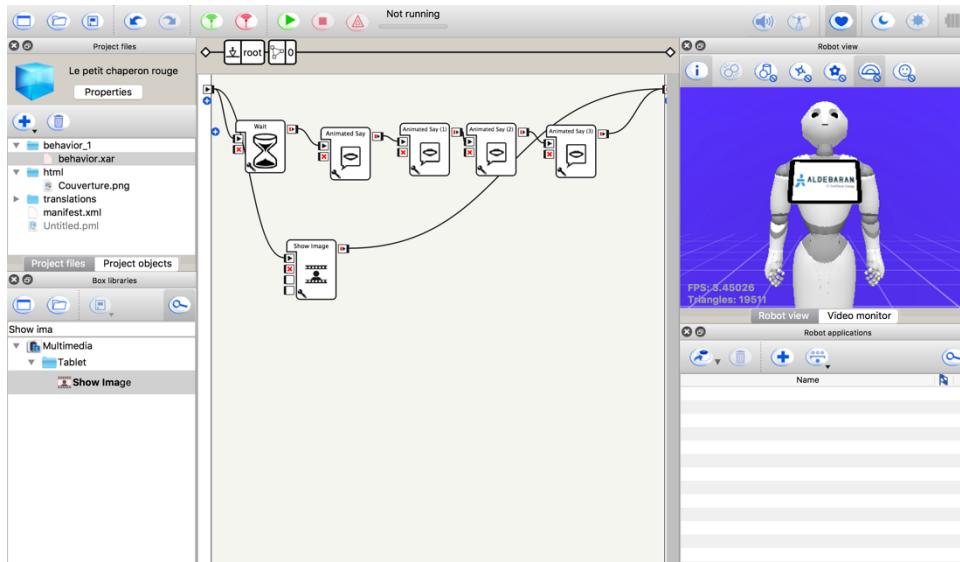


15. Sélectionnez l'image, qui doit être au format *png* et appuyez sur *Open*. L'image apparaît dans la fenêtre *Project files* mais au mauvais endroit.

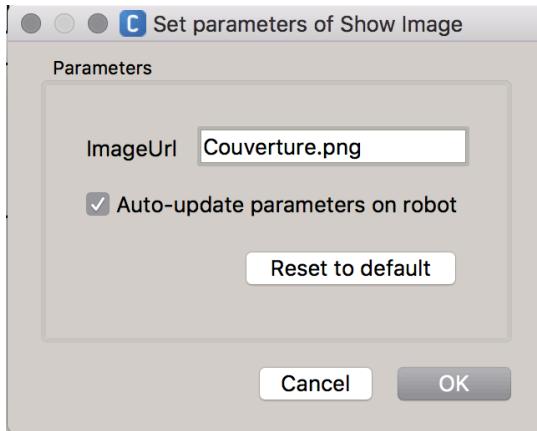
16. Placez le fichier dans le dossier *html*.



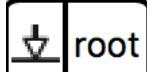
17. Ajoutez la boîte *Show image* dans la boîte *Diagram* et à la connecter parallèlement aux boîtes *Wait* et *Animated Say*. Vous la trouverez dans le dossier MULTIMEDIA, dans le sous-dossier TABLET.



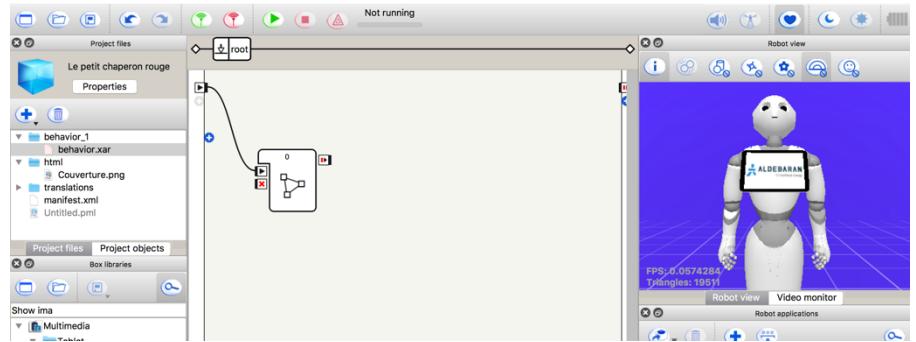
18. Cliquez sur la clé à molette de la boîte *Show image* pour pouvoir ajouter le nom de l'image que vous souhaitez afficher lors de la lecture de cette boîte.



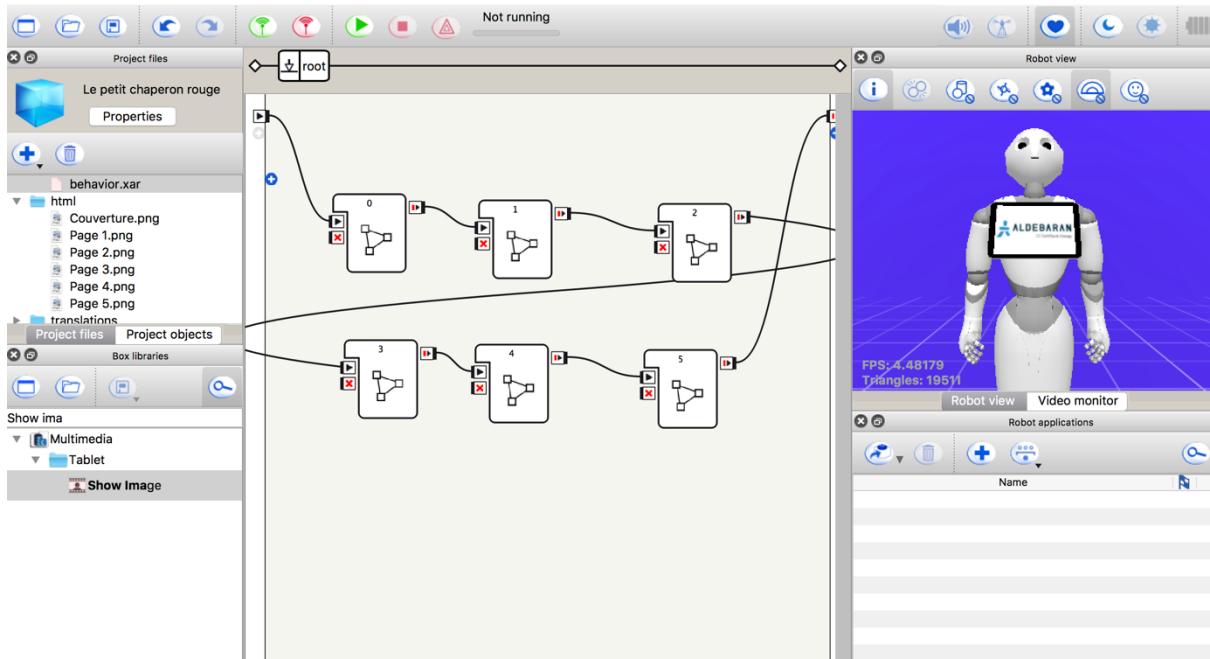
Dans la boîte 0, nous souhaitons que PEPPER affiche l'image de la couverture sur sa tablette.



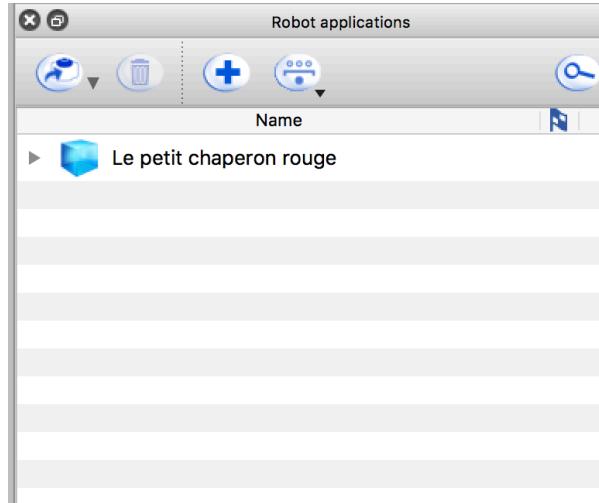
19. Appuyez sur **root** pour revenir à l'espace de travail principal. Vous venez de terminer de programmer votre première boîte *Diagram*, qui est, dans cet exemple, l'introduction du livre.



20. Maintenant pour chaque page, vous devez répéter les étapes 6 à 18, c'est-à-dire, créer une nouvelle boîte *Diagram*, ajouter une boîte *Say* au début du programme, paramétrer plusieurs boîtes *Animated Say* et afficher une image. A la fin, votre programme ressemblera à cela :



21. Vous pouvez enregistrer le programme dans le robot en appuyant sur . Le titre du programme apparaît dans la liste des applications et sur la tablette du robot.

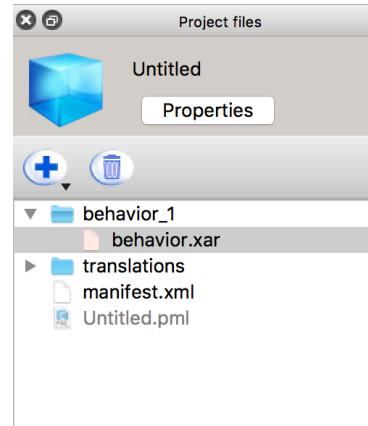


## 4.2. Jeu de quilles

Ce programme permet à PEPPER de jouer aux quilles, c'est-à-dire qu'il va lancer une balle et essayer de faire tomber le maximum de quilles.

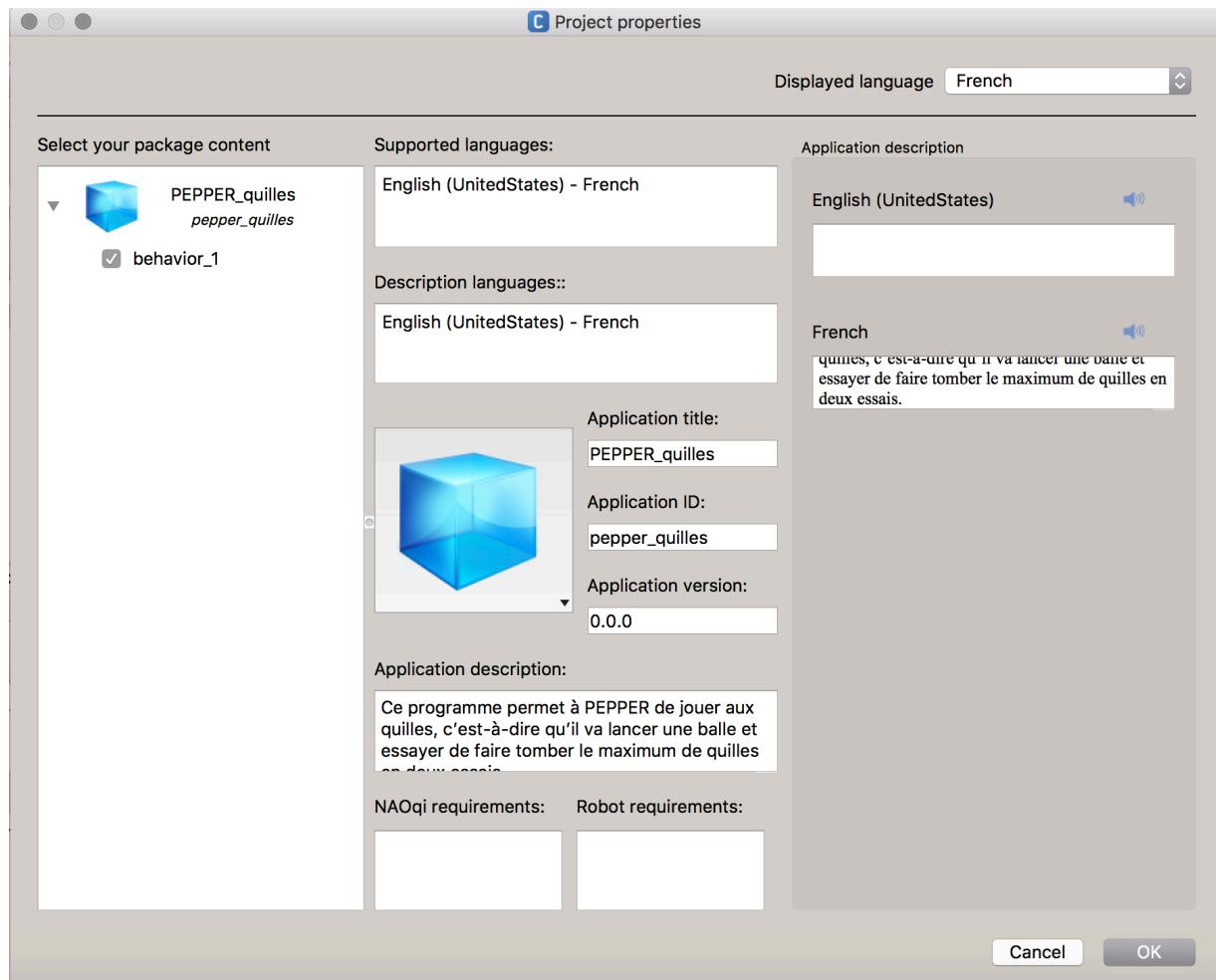
1. Ouvrez Choregraphe

2. Cliquez sur *Properties* dans la fenêtre *Project files*

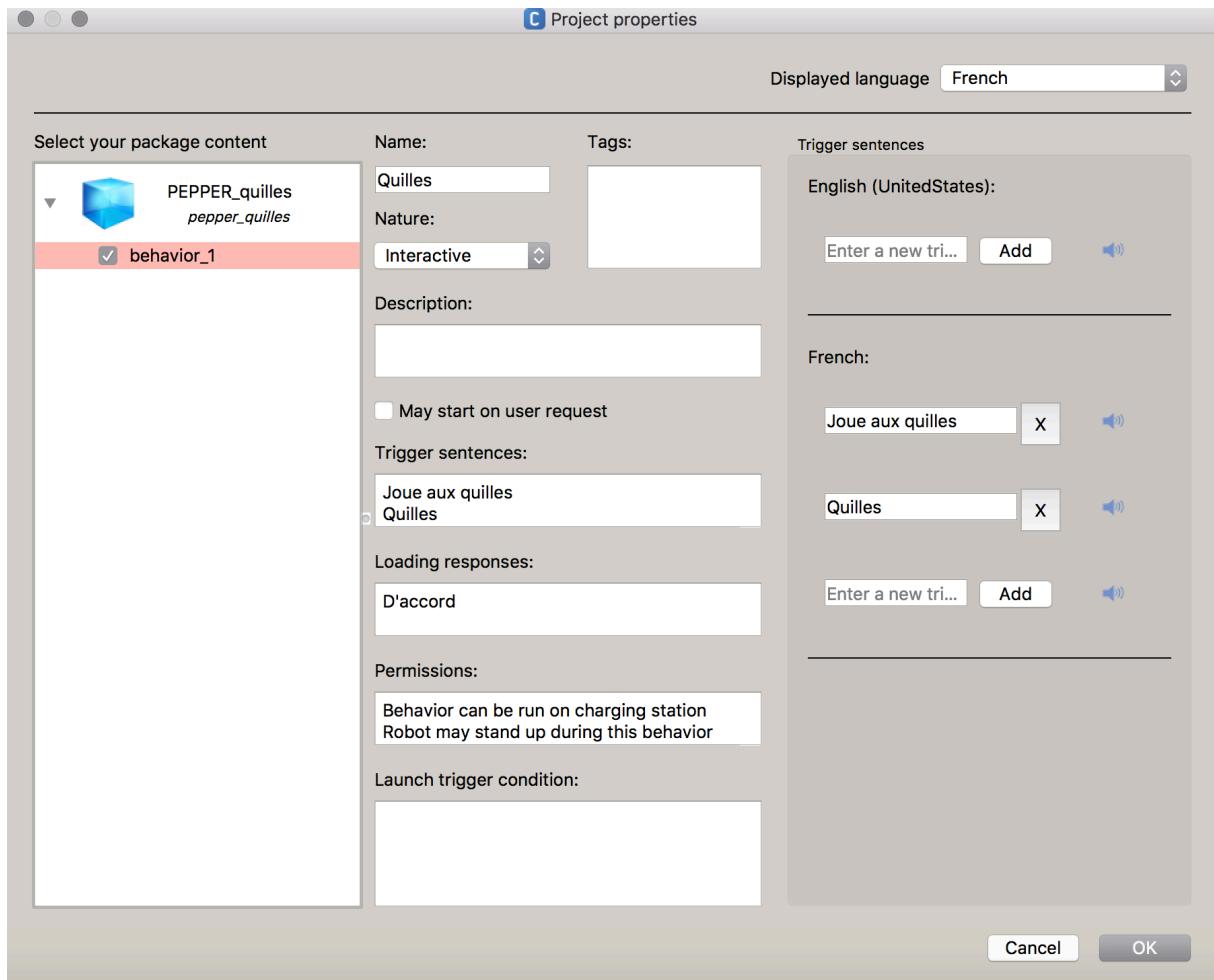


3. Définissez les propriétés de votre projet : Choisissez la langue, mettez un titre, une description et une image. L'image apparaîtra sur la tablette de PEPPER.

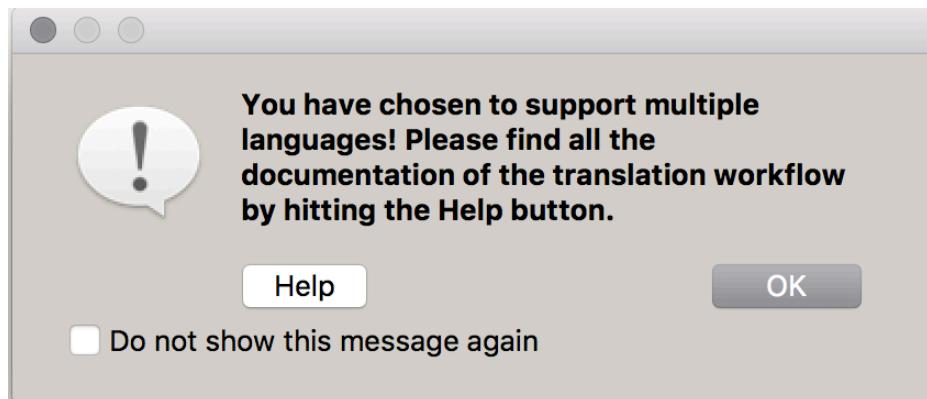
Vous êtes obligé de recopier votre titre français dans l'onglet *English (UnitedStates)* pour le voir apparaître comme titre du fichier sur l'interface de Choregraphe et dans la tête du robot.



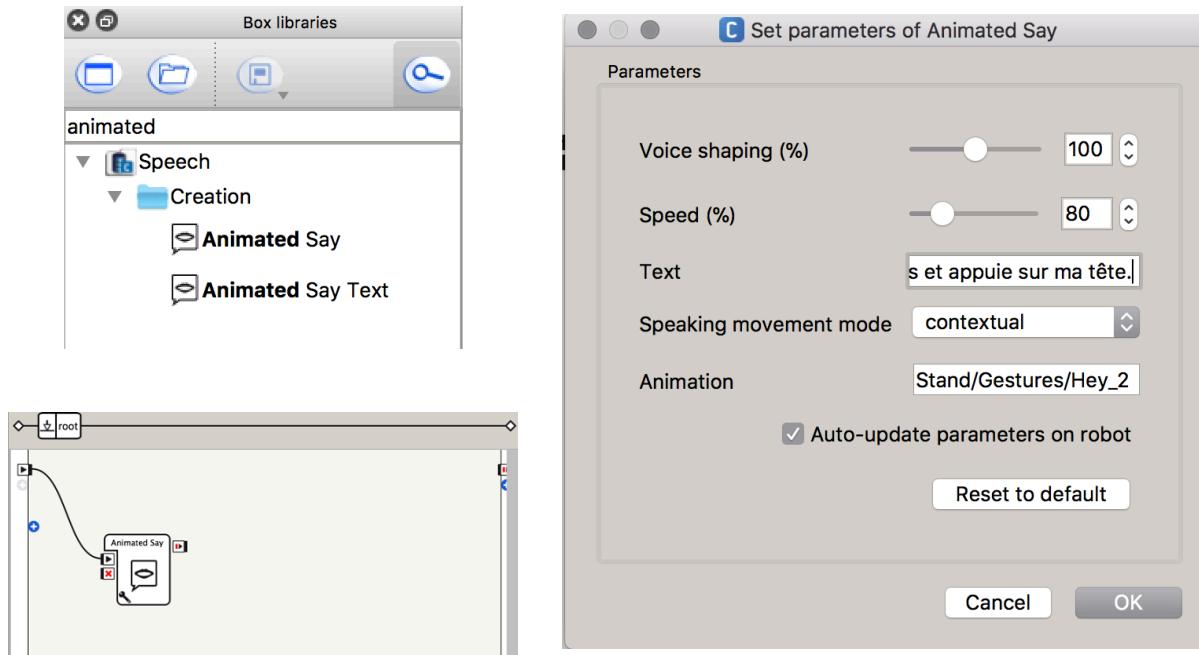
4. Cliquez sur *behavior\_1* les paramètres du comportement : Nommez le, choisissez sa nature, créez des phrases de déclenchement, des réponses et donnez des permissions.



5. Cliquez sur Ok

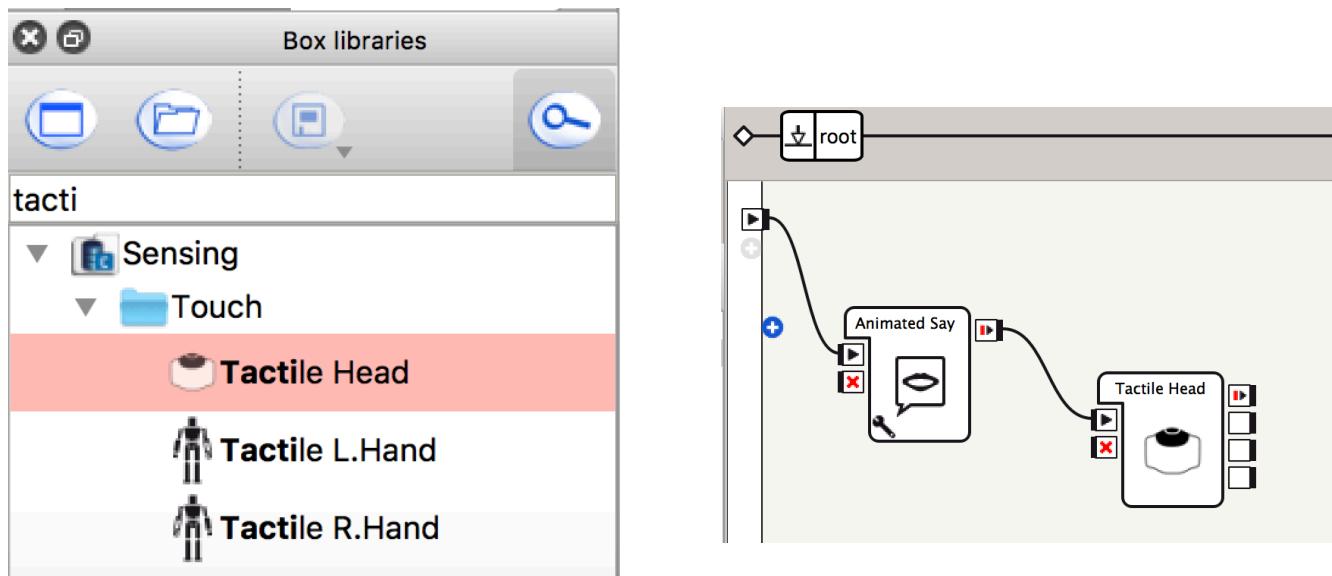


6. Commencez par ajouter une boîte *Animated Say* pour donner une consigne. Cliquez sur la clé à molette  pour définir les paramètres de la boîte.

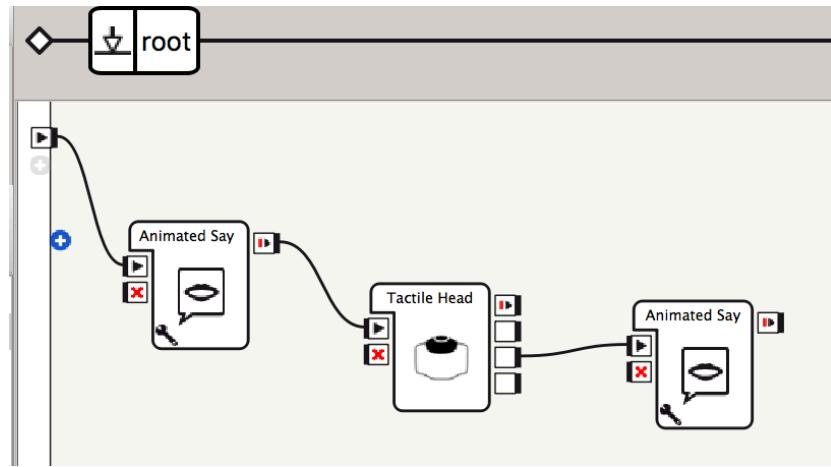


Voice le texte que vous pouvez écrire : « Place moi devant les quilles et appuie sur ma tête. »

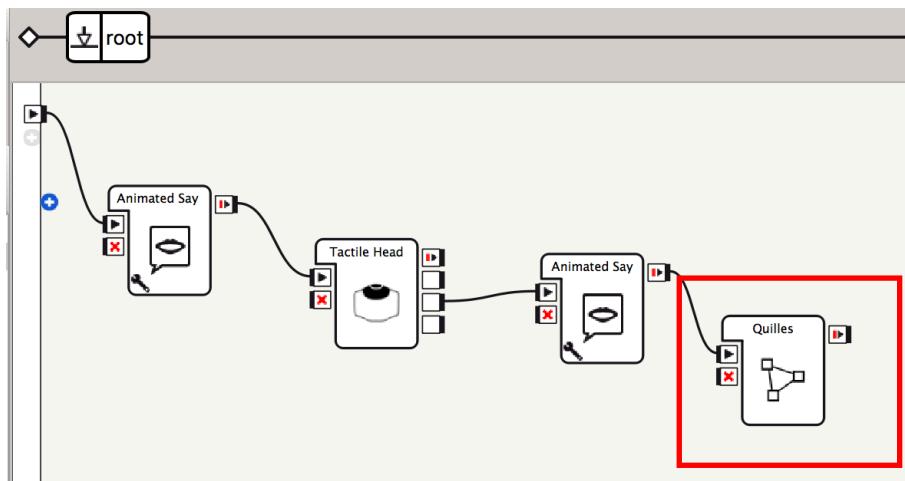
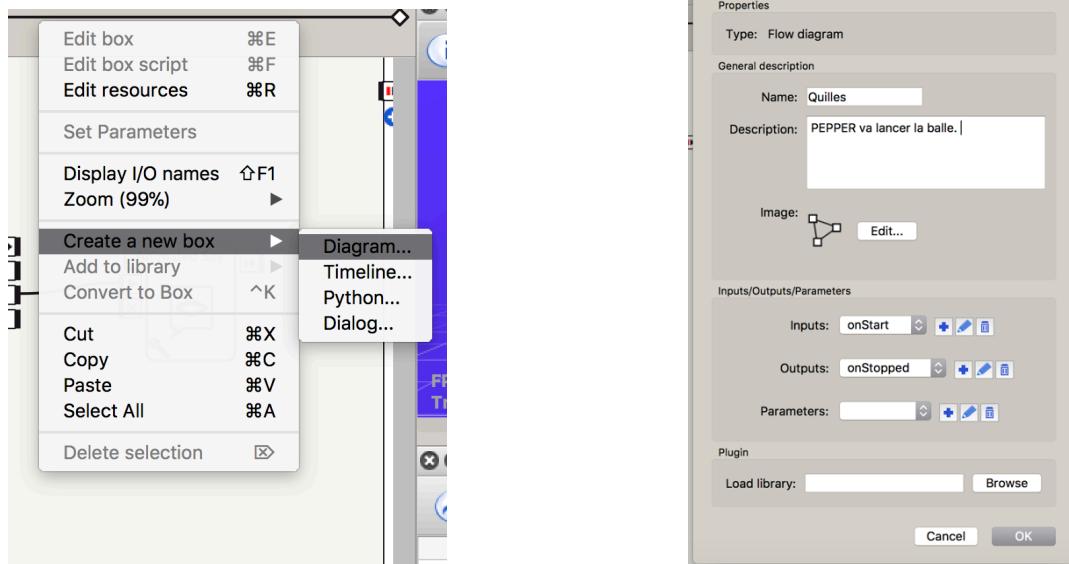
7. Ajoutez une boîte *Tactile Head*.



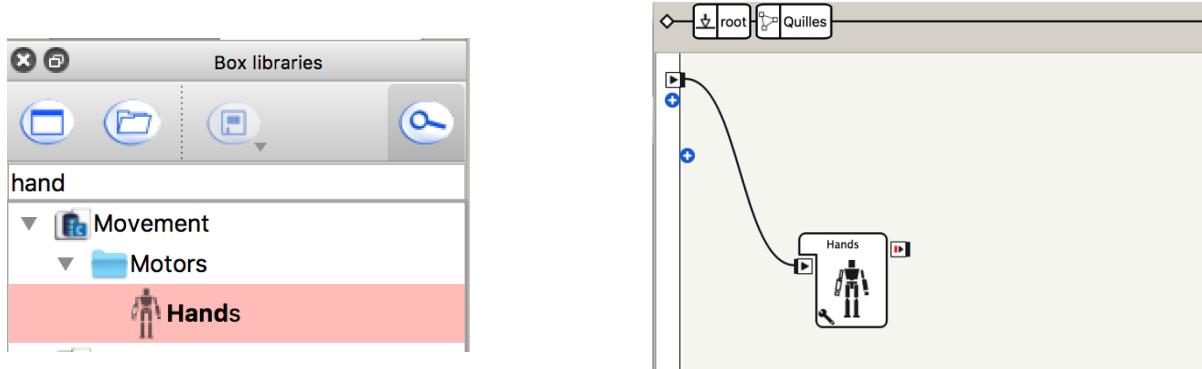
8. Ajoutez une boîte *Animated Say* pour donner une nouvelle consigne qui pourrait être la suivante : « Donne-moi la balle dans ma main droite et appuie derrière ma main. »



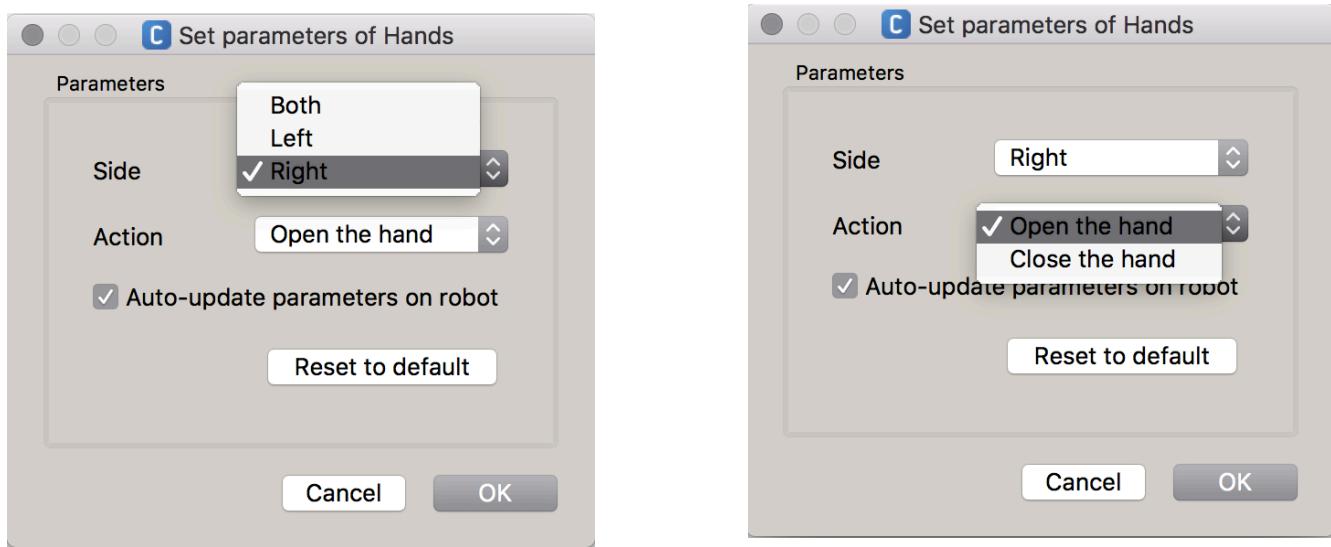
9. Créez une boîte *Diagram* nommée *Quilles*.



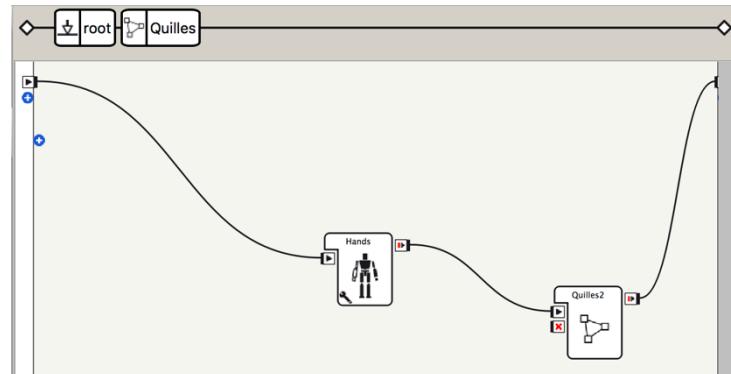
10. Double-cliquez dessus pour y entrer et ajoutez la boîte *Hands* pour que PEPPER ouvre la main droite.



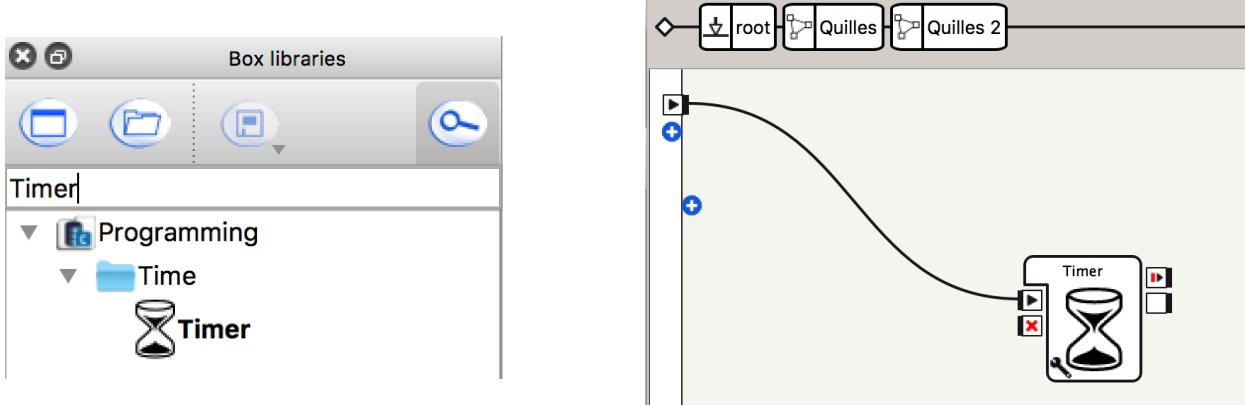
11. Cliquez sur la clé à molette pour définir les paramètres de la boîte *Hands*. Sélectionnez *Right* et *Open the hand*.



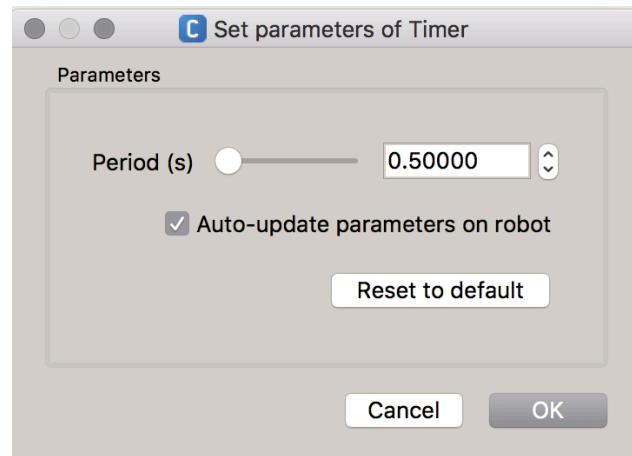
12. Créez une nouvelle boîte *Diagram* intégrée à la boîte *Quilles*. Appelez la *Quilles2*. Connectez la tout de suite à l'OnStopped



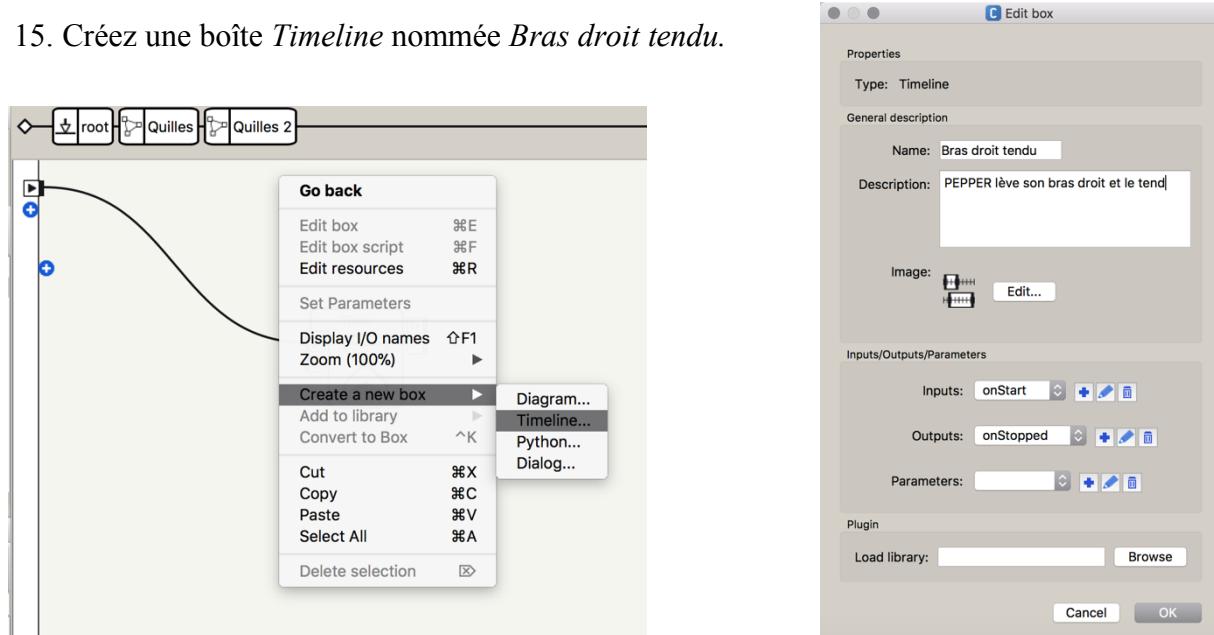
13. Entrez dans cette nouvelle boîte et ajoutez-y une boîte *Timer*. Elle permet d'envoyer une infinité de signaux réguliers.



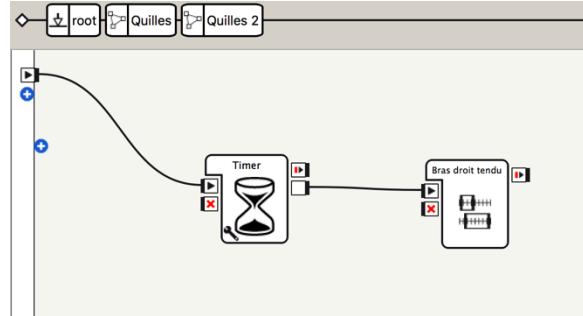
14. Cliquez sur la clé à molette pour définir la période et mettez-en une très petite pour que le signal soit continu.



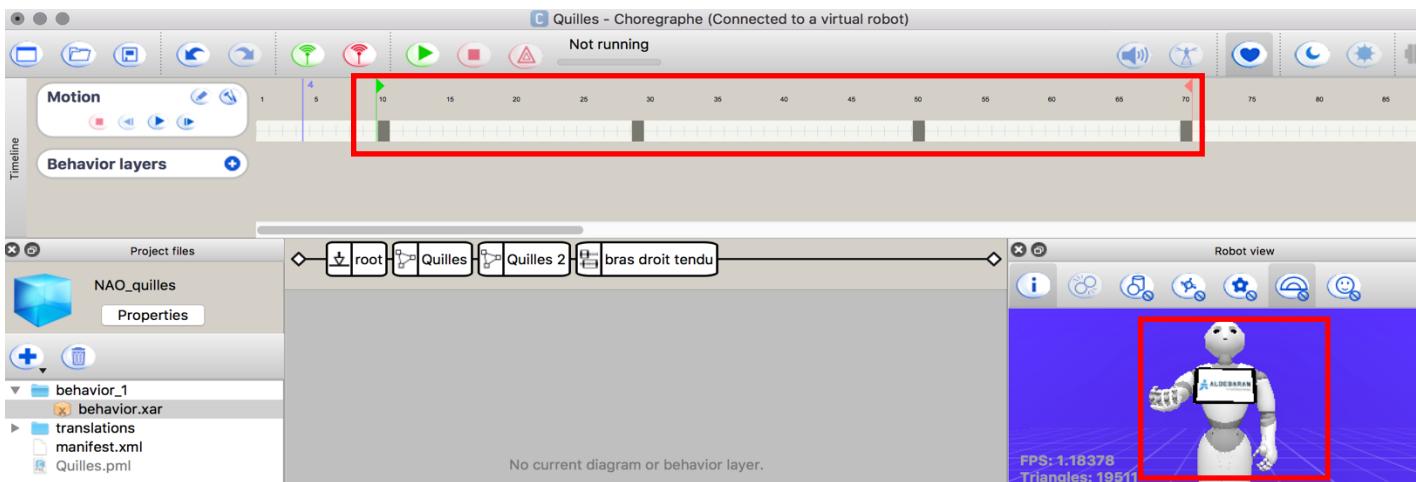
15. Créez une boîte *Timeline* nommée *Bras droit tendu*.



14. Connectez la à la boîte *Timer* via la sortie du bas qui permet d'envoyer le signal à intervalle régulier.

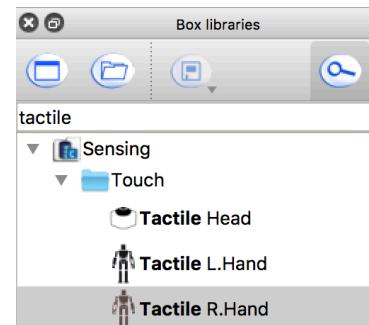


15. Double-cliquez dessus et placez plusieurs marqueurs sur la frise de temps où PEPPEL a le bras droit tendu, paume de la main vers le haut. La position doit être la même pour tous les marqueurs. Pensez à bien délimiter vos positions. Le point de départ doit être au niveau du premier marqueur et le point final doit être au niveau du dernier. Il est conseillé d'utiliser le vrai robot pour créer les positions.

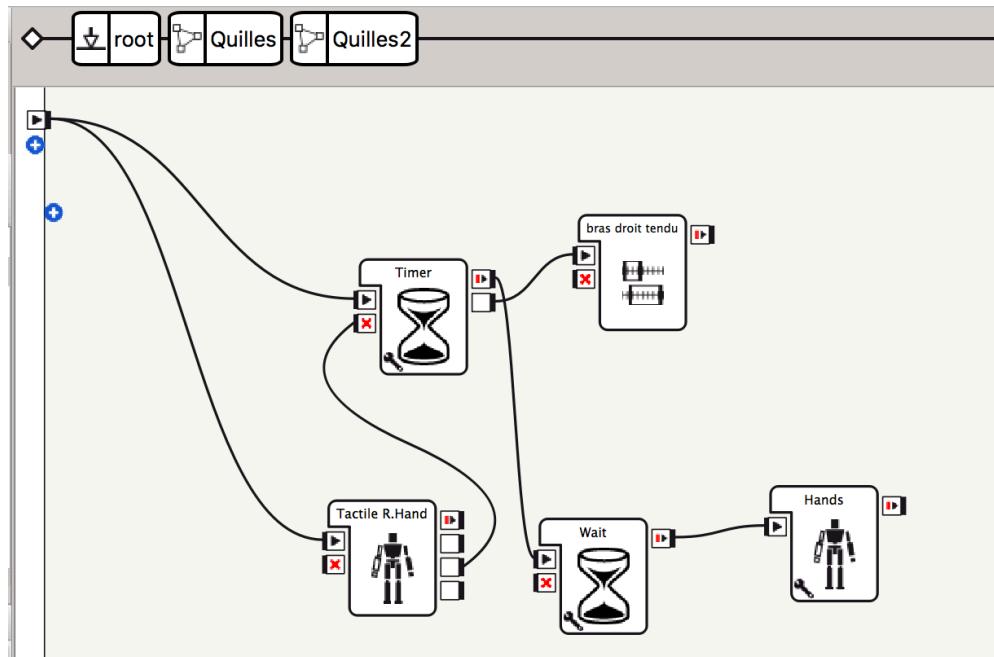


Arrivé à ce stade du programme, PEPPEL reste statique, bras droit tendu, paume de la main ouverte vers le ciel. Il attend que vous lui donnez une balle.

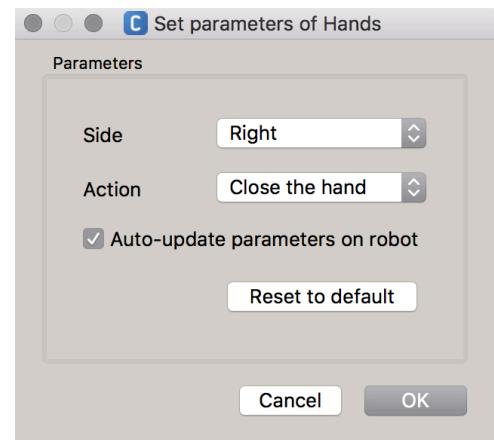
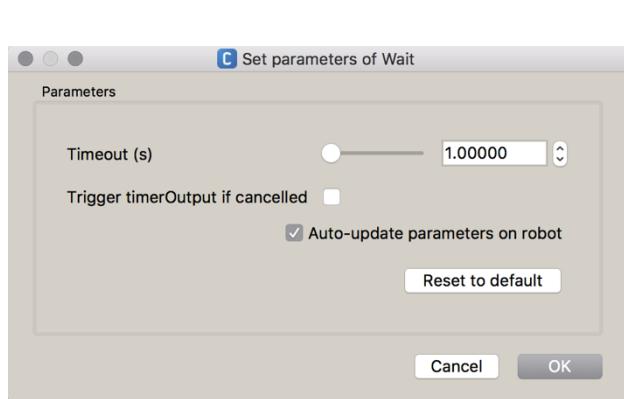
15. Maintenant, il va devoir fermer la main pour maintenir la balle. Pour cela ajoutez une boîte *Tactile R. Hand* qui vous permettra, au contact du capteur, de demander à PEPPEL de replier ses doigts et de se préparer à lancer. Connectez la à l'*OnStart* et à l'entrée *onStop* de la boîte *Timer*. Le signal à répétition va s'arrêter et donc le bras droit ne va pas plus rester statique.



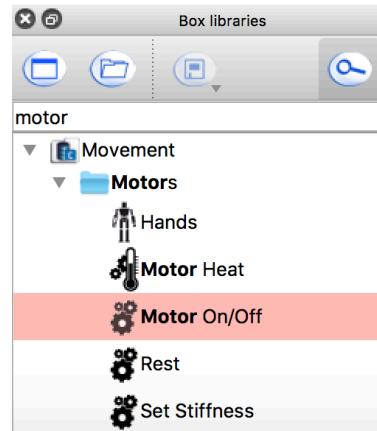
16. Ajoutez une boîte *Wait* et à une boîte *Hands* dans votre espace de travail de la boîte *Quilles2* et connectez les à la sortie *Onstopped* de la boîte *Timer*.

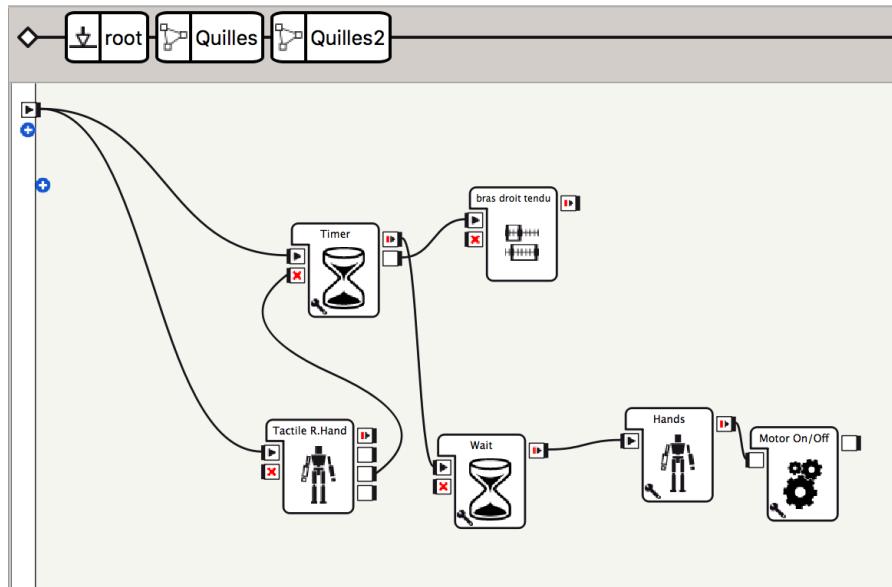


Pour les paramétriser appuyez sur les clés à molette . Définissez une seconde d'attente pour la boîte *Wait*. Choisissez *Right* et *Close the hand* pour la boîte *Hands*.

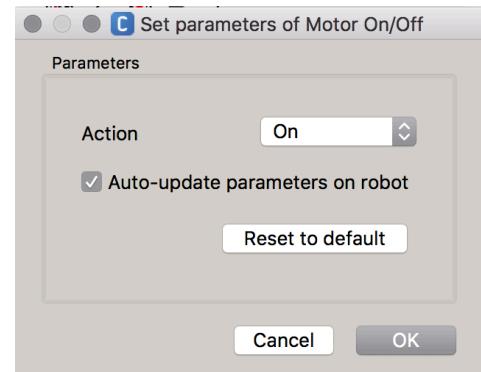


17. Ajoutez une boîte *Motors*. Cette boîte permet de serrer les moteurs et va donc maintenir la main droite fermée lorsque vous allez créer un mouvement.



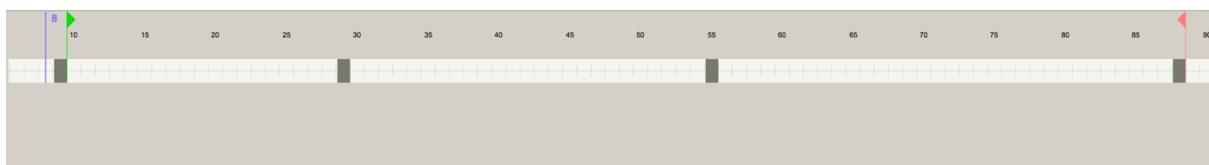
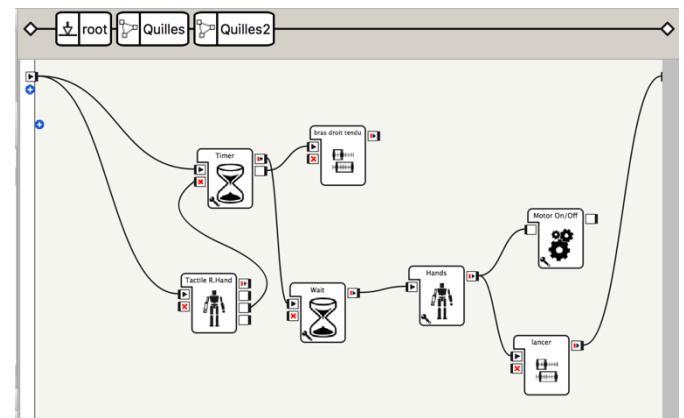


18. Cliquez sur la clé à molette et sélectionnez *On*.

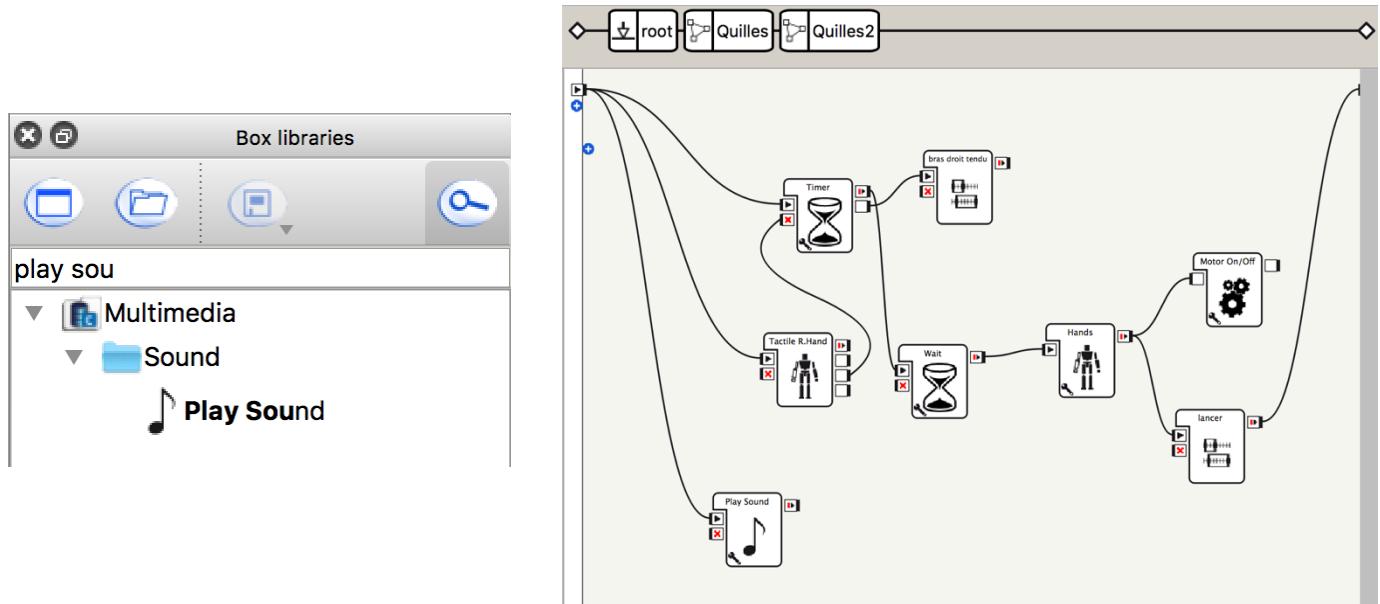


19. Créez une nouvelle boîte *Timeline* appelée *Lancer* et connectez la également à la boîte *Hands*.

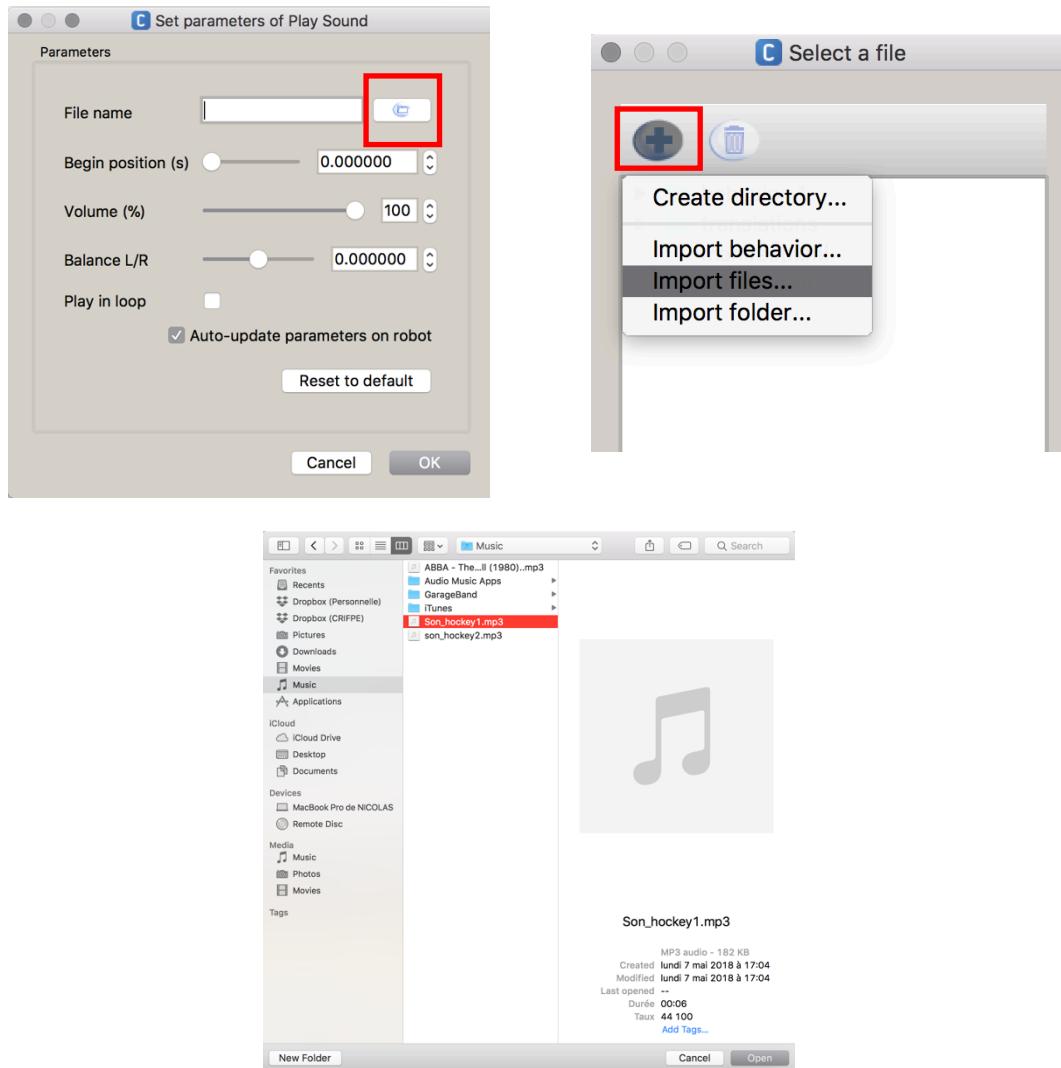
20. Double-cliquez dessus et créez un mouvement de lancer, avec le robot réel. Enregistrez bien chaque position sur la frise de temps. Il vous faut au moins une position en bas, une position où la main est bien en face des quilles et une position haute. N'oubliez pas sur la position haute de desserrer les doigts pour que PEPPER puisse lâcher la balle.



21. Vous pouvez intégrer un son au début du lancer. Pour cela, ajouter la boîte *Play Sound*.

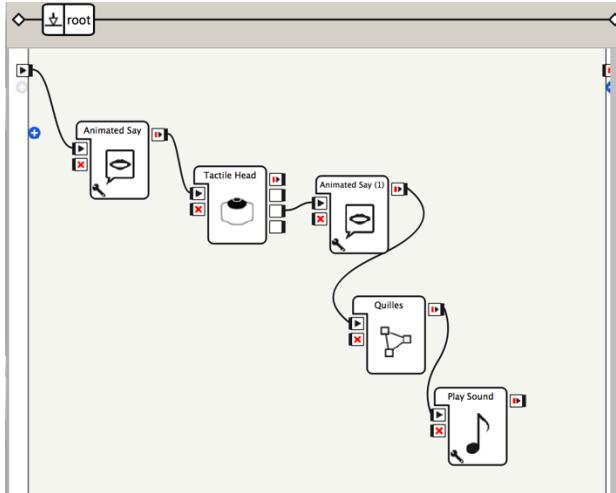


Cliquez sur la clé à molette et choisissez votre son.

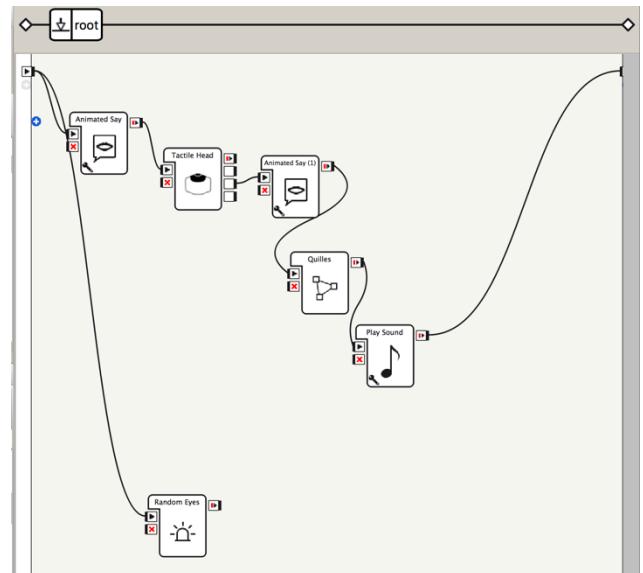
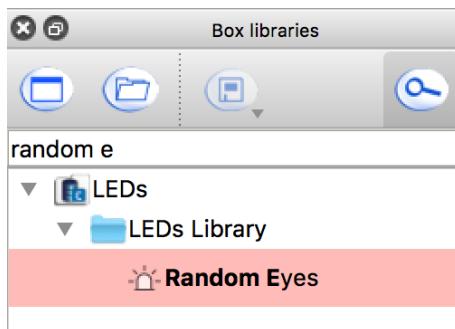


 root

22. Cliquez sur  pour revenir à l'espace de travail principal. Ajoutez un nouveau son pour célébrer le lancer.



23. Vous pouvez ajouter la boîte *Random Eyes* si vous souhaitez que PEPPER ait les yeux qui changent de couleur.



24. Le programme est terminé, vous pouvez enregistrer le programme dans le robot en appuyant sur . Le titre du programme apparaît dans la liste des applications et sur la tablette du robot.



Ce programme est une base pour que PEPPER jouer aux quilles mais il y a plusieurs manières de l'améliorer : proposer un deuxième lancer, créer une célébration de fin ... A vous de faire des tests et de le rendre divertissant.

### 4.3. Jeu des capteurs

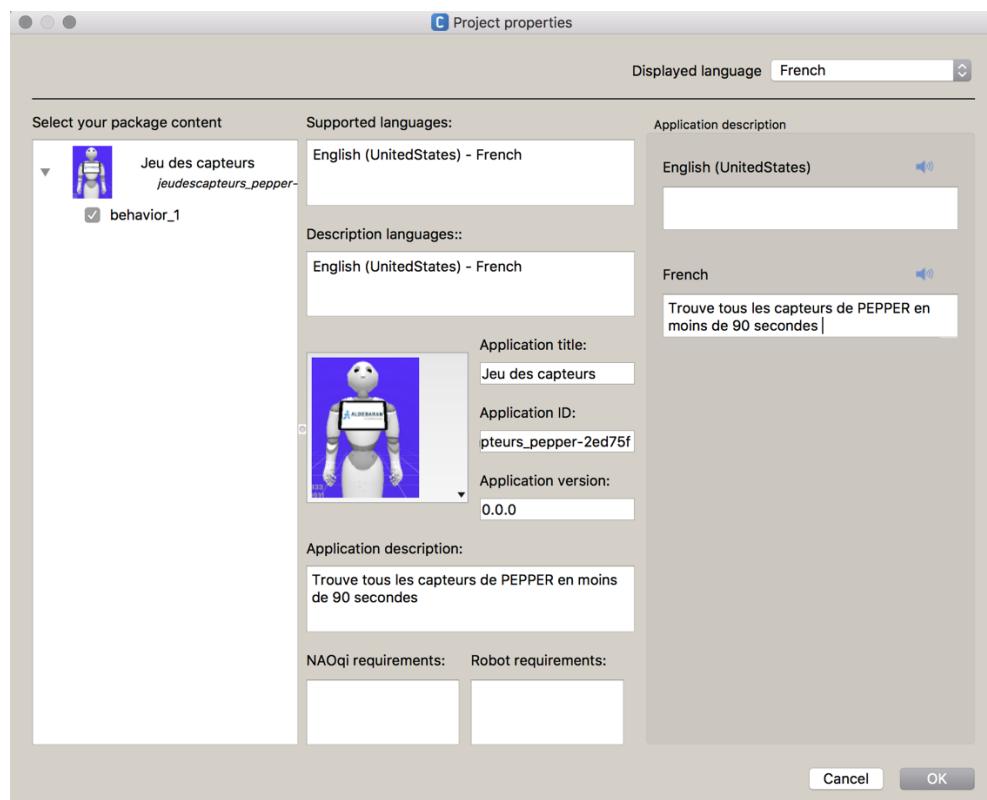
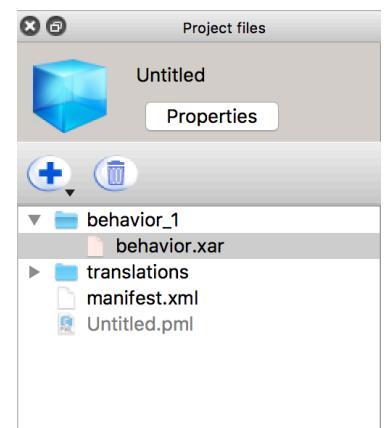
Le but de ce jeu est de trouver tous les capteurs de PEPPER. L'humain a 90 secondes pour toucher l'ensemble des capteurs du robot mise à part sa tablette. S'il réussit, PEPPER le félicite et s'il échoue, PEPPER l'encourage à réessayer plus tard.

1. Ouvrez Choregraphe

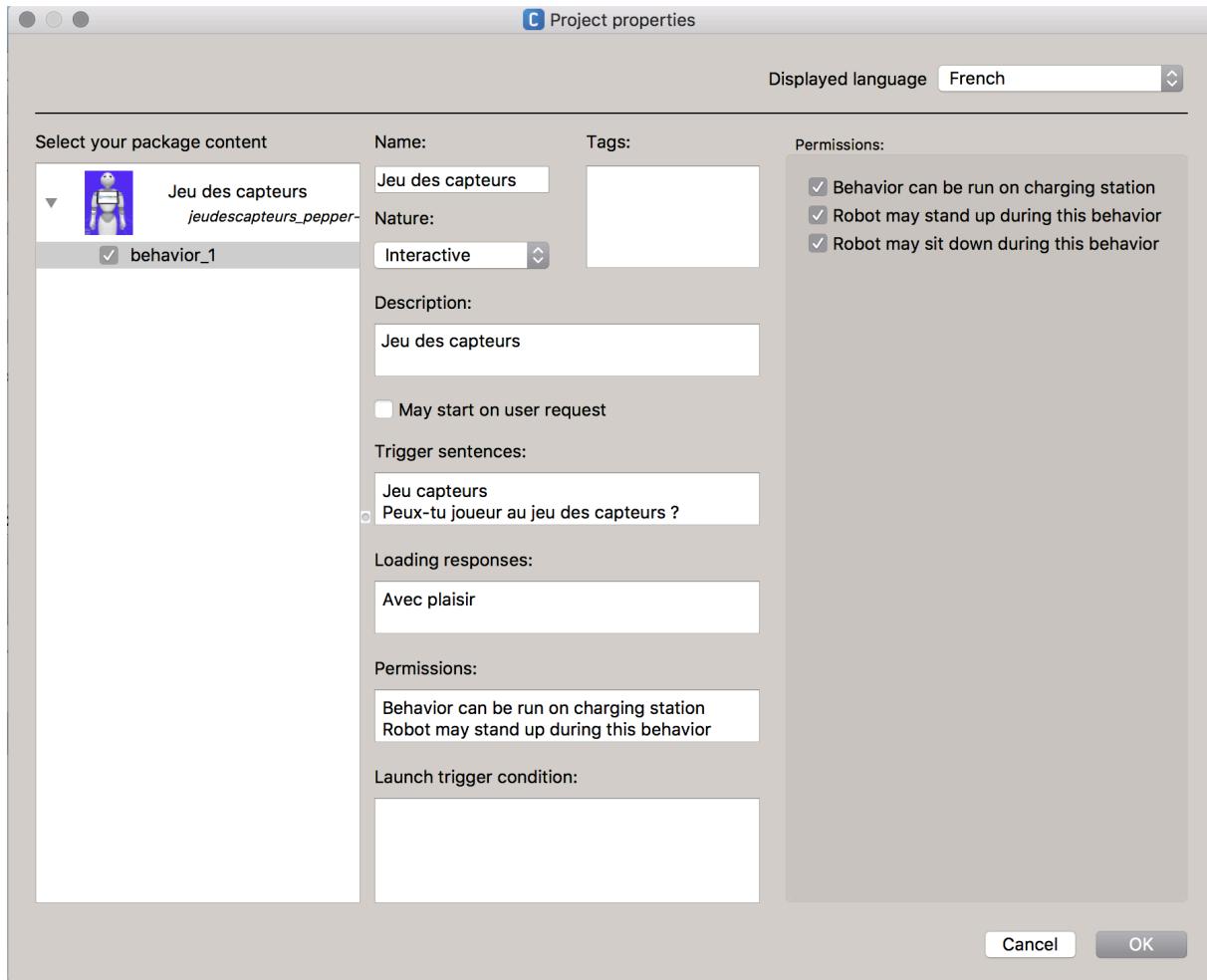
2. Cliquez sur *Properties* dans la fenêtre *Project files*

3. Définissez les propriétés de votre projet : Choisissez la langue, mettez un titre, une description et une image. L'image apparaîtra sur la tablette de PEPPER.

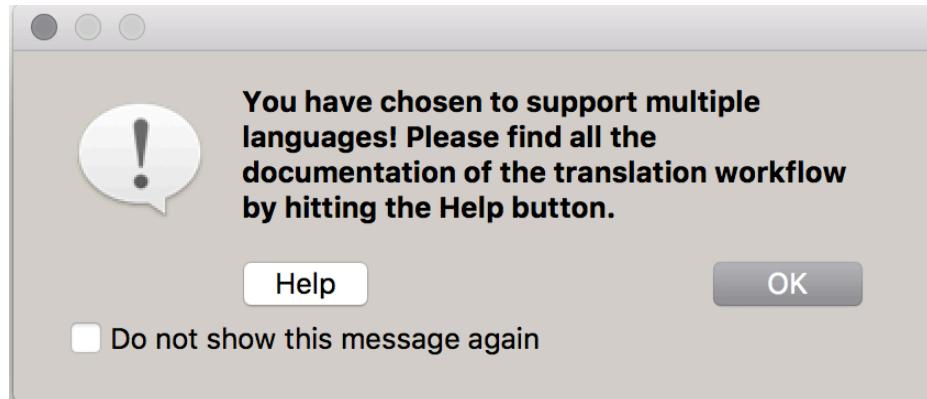
Vous êtes obligé de recopier votre titre français dans l'onglet *English (UnitedStates)* pour le voir apparaître comme titre du fichier sur l'interface de Choregraphe et dans la tête du robot.



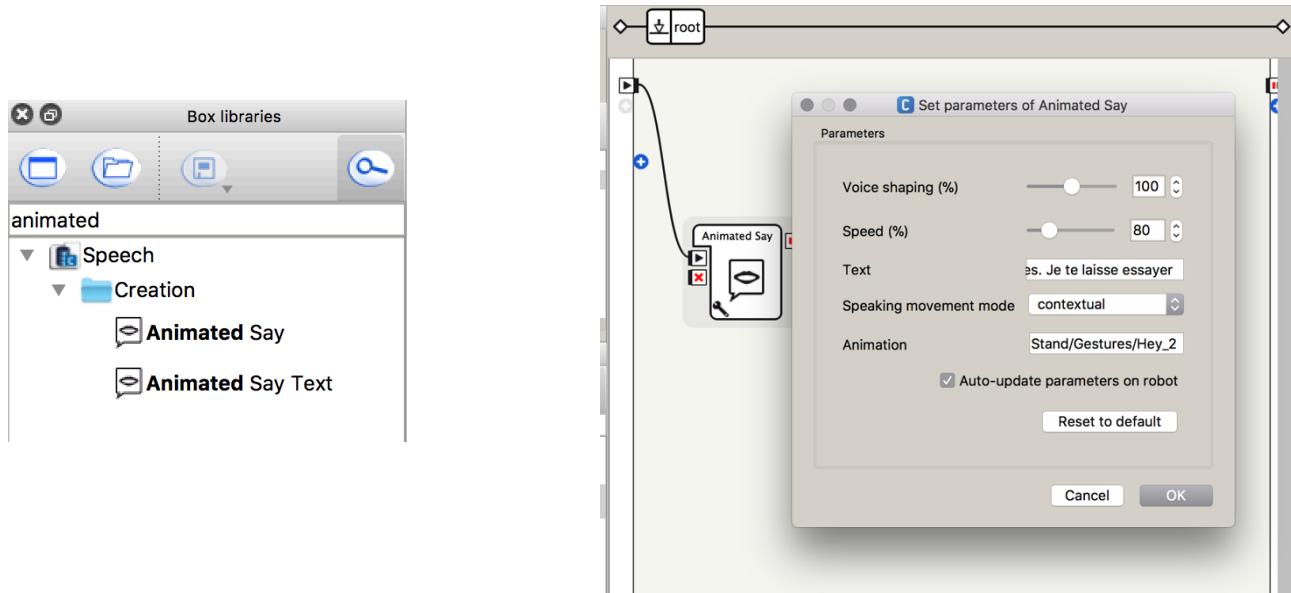
4. Cliquez sur *behavior\_1* les paramètres du comportement : Nommez le, choisissez sa nature, créez des phrases de déclenchement, des réponses et donnez des permissions.



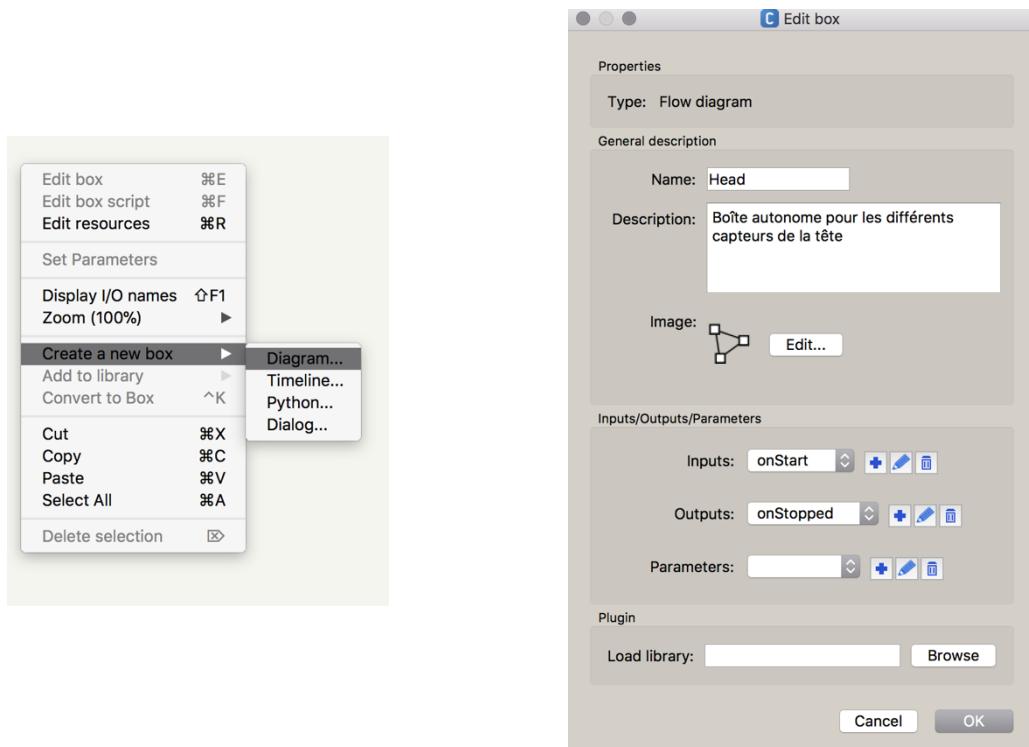
5. Cliquez sur Ok



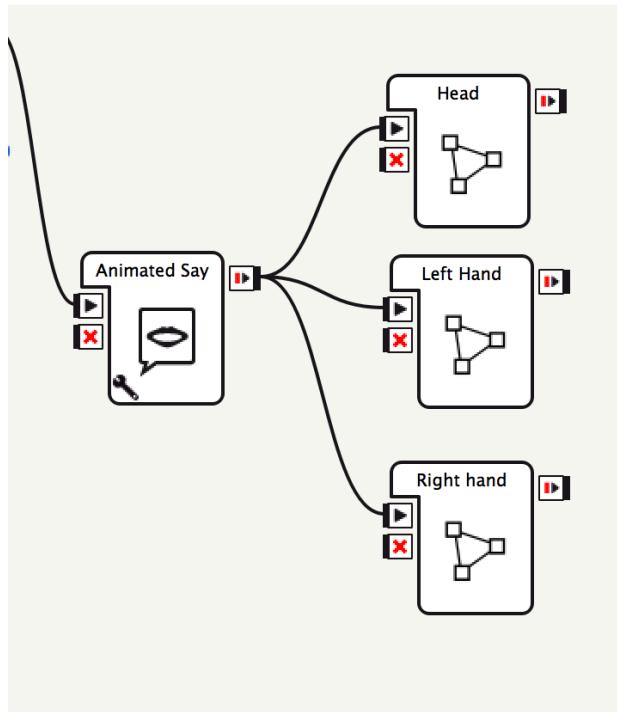
6. Commencez par ajouter une boîte *Animated Say* pour expliquer les règles du jeu. Cliquez sur la clé à molette  pour définir les paramètres de la boîte.



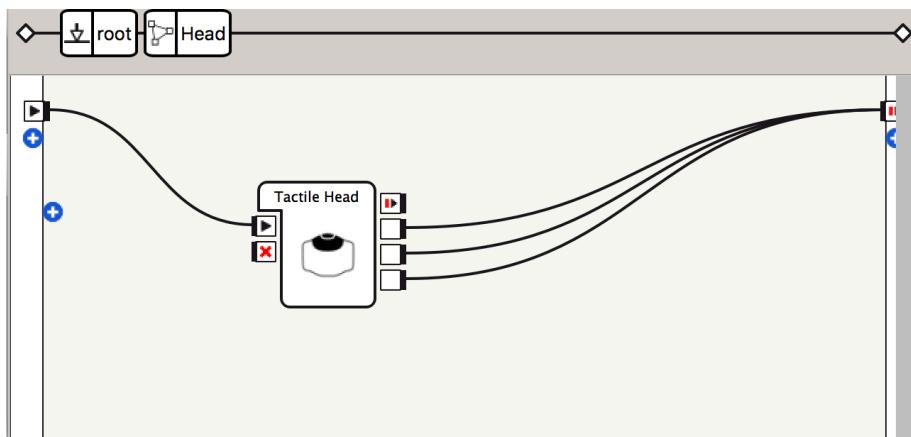
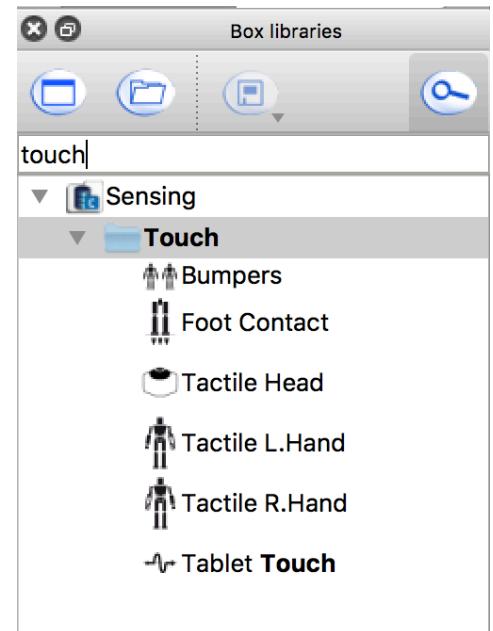
7. Ensuite, créez une boîte *Diagram* pour la tête, la main gauche et la main droite puisqu'ils comportent, chacun, 3 capteurs différents mais vous souhaitez que la personne en touche seulement un. Pour cela, cliquez droit dans votre espace de travail, puis appuyez sur *Create a new box* et *Diagram*. Définissez les propriétés.



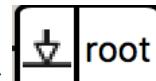
Voilà à quoi cela doit ressembler.



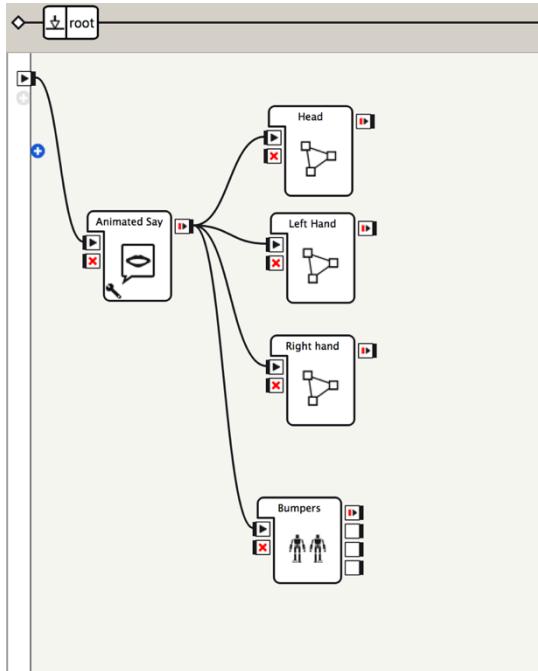
8. Ensuite, double-cliquez sur une des boîtes que vous venez de créer et insérez la boîte du capteur correspondant. Chaque sortie des boîtes représente un capteur différent, mise à part la sortie la plus haute avec un triangle à l'intérieur.



9. Répétez l'opération pour les autres parties du corps.

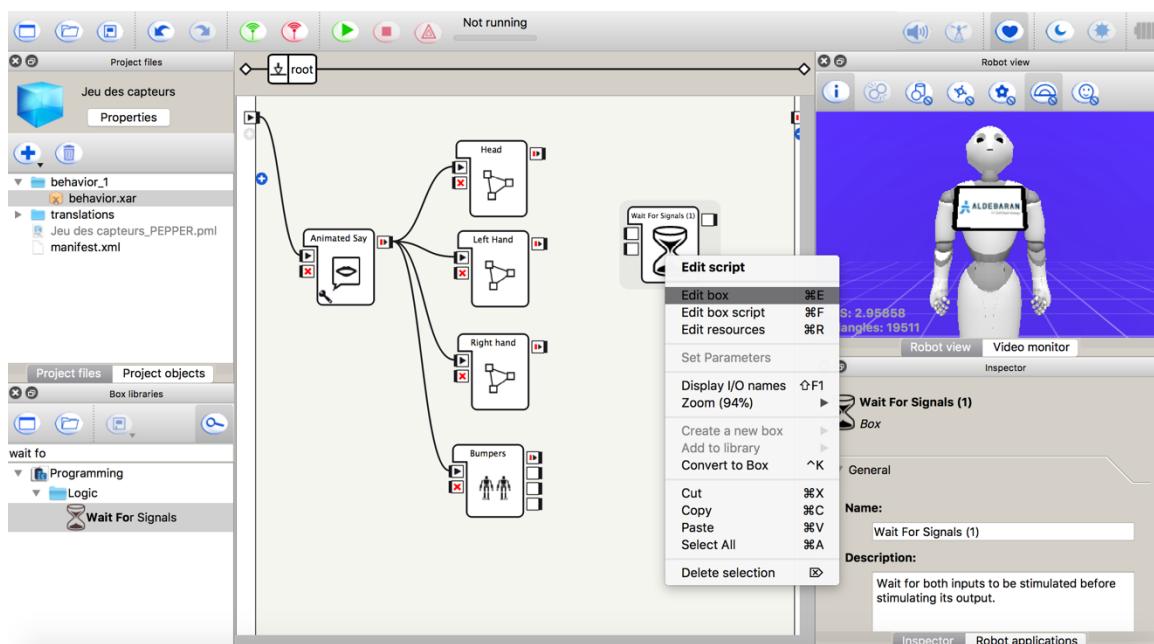


10. Revenez dans votre espace de travail principal en appuyant sur **root** et ajoutez la boîte *Bumpers* pour les capteurs des pieds.



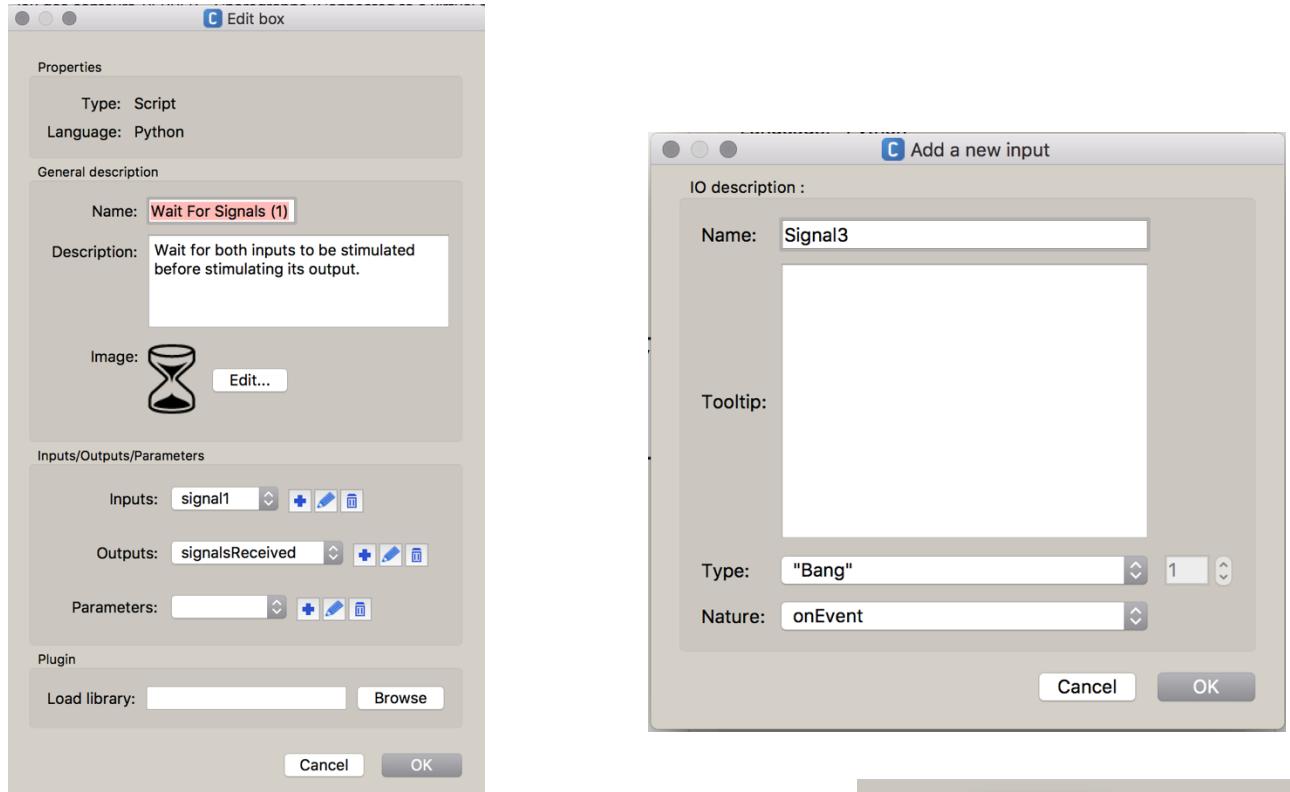
Vous avez donc maintenant 4 boîtes de capteur, une pour la tête, une pour la main gauche, une autre pour la main droite et un dernier pour les pieds.

11. Maintenant ajoutez la boîte *Wait for signal* dans votre espace de travail. Cette boîte permet d'attendre l'ensemble des signaux demandés pour envoyer un signal. Cliquez droit dessus et appuyer sur *Edit box*. Vous allez ajouter des entrées pour que chaque capteur ait la sienne.

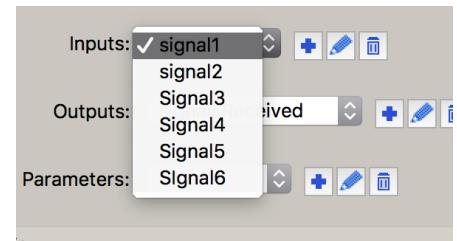




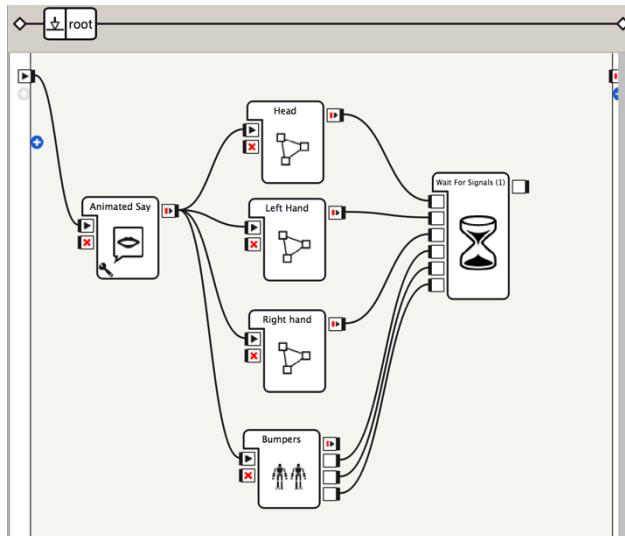
12. Cliquez sur le à côté de la ligne *Inputs*. Nommez le nouveau signal et cliquez sur *Ok*. Renouvelez l'opération autant de fois que vous avez des capteurs.



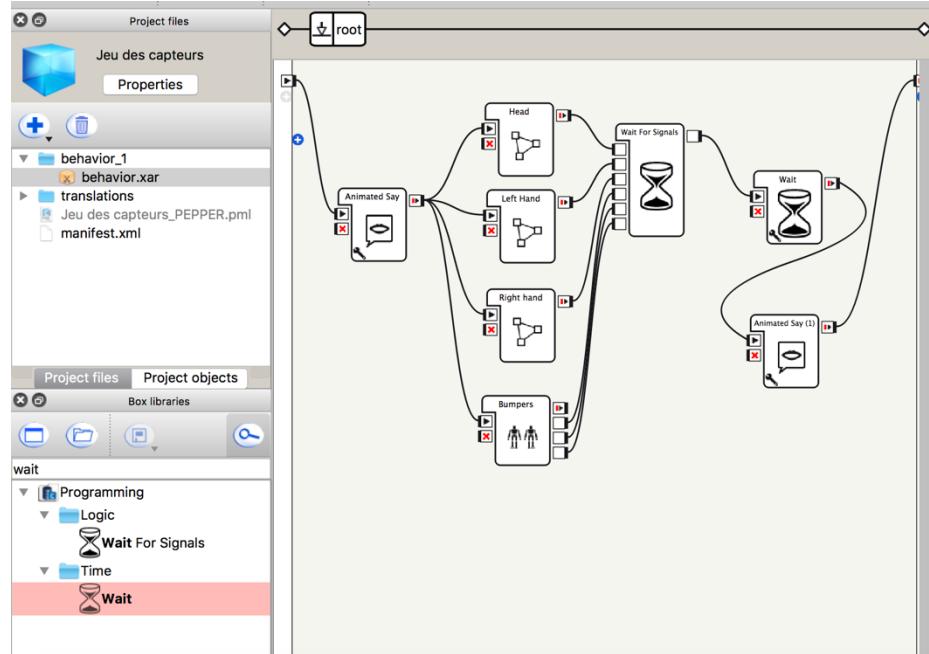
A la fin vous devez apercevoir l'ensemble des entrées en cliquant sur les flèches de la ligne *Inputs*.



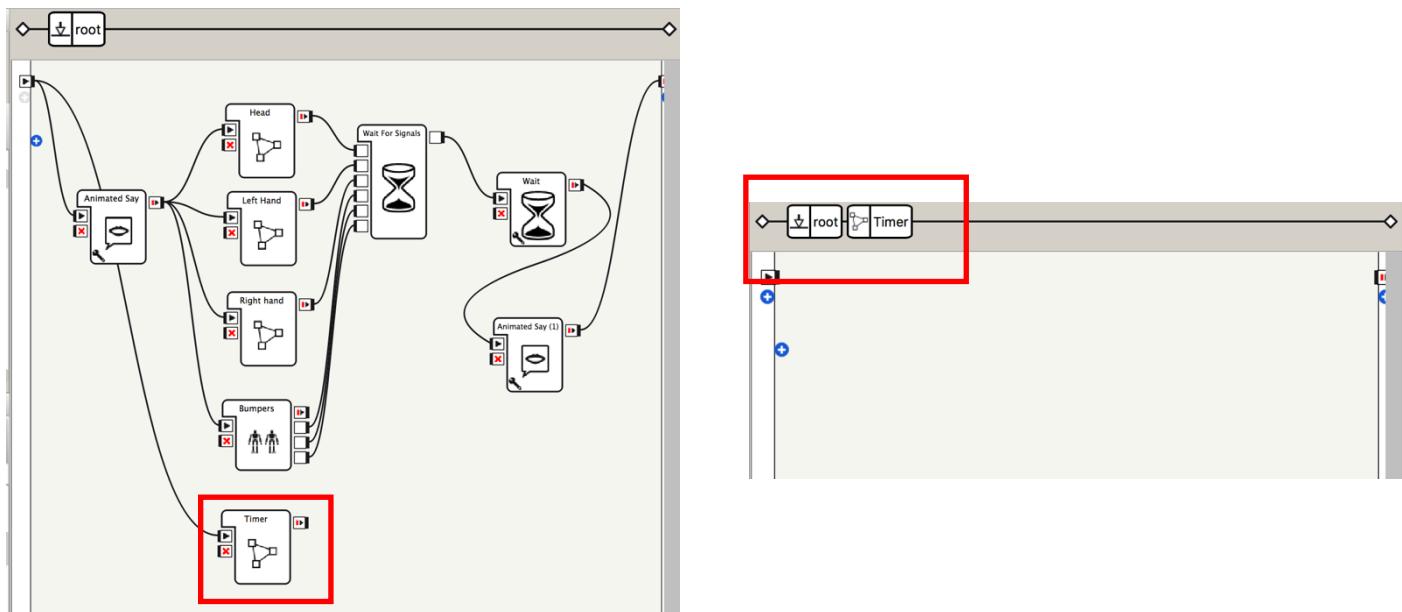
13. Reliez tous les signaux de capteurs à la boîte *Wait for signal*



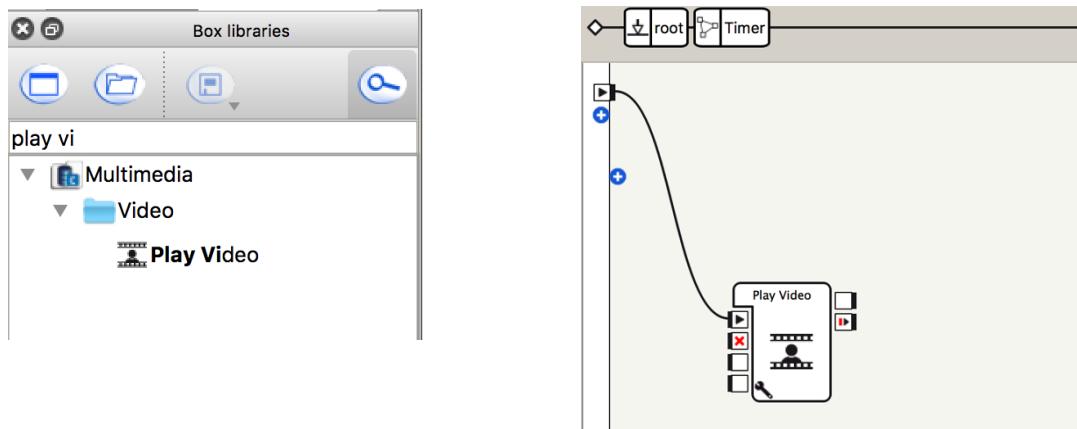
14. Ensuite, ajoutez une boîte *Way* une boîte *Animated Say* pour clôturer le jeu. La boîte *Way* permet de marquer un blanc avant que le robot parle pour féliciter le joueur. Vous pouvez définir le temps d'attente en cliquant sur .



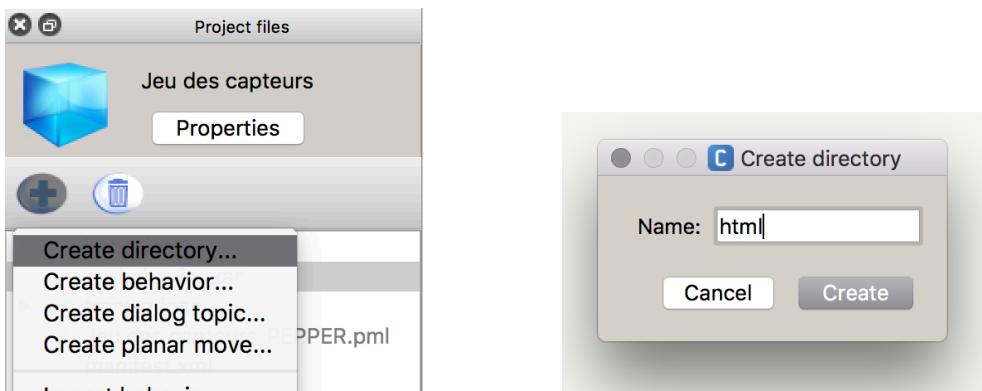
15. Maintenant vous devez créer une boîte *Diagram* que vous allez programmer en tant que compte à rebours. Nommez la comme vous le souhaitez, puis double-cliquez dessus.



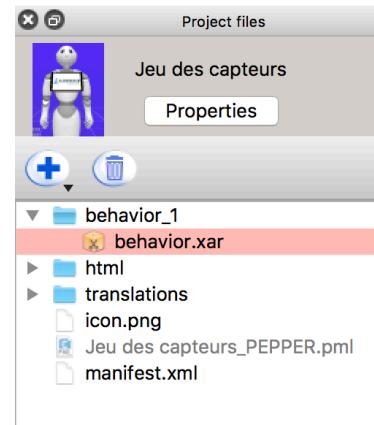
16. Ajoutez la boîte *Play Video* dans la boîte *Diagram* que vous venez de créer.



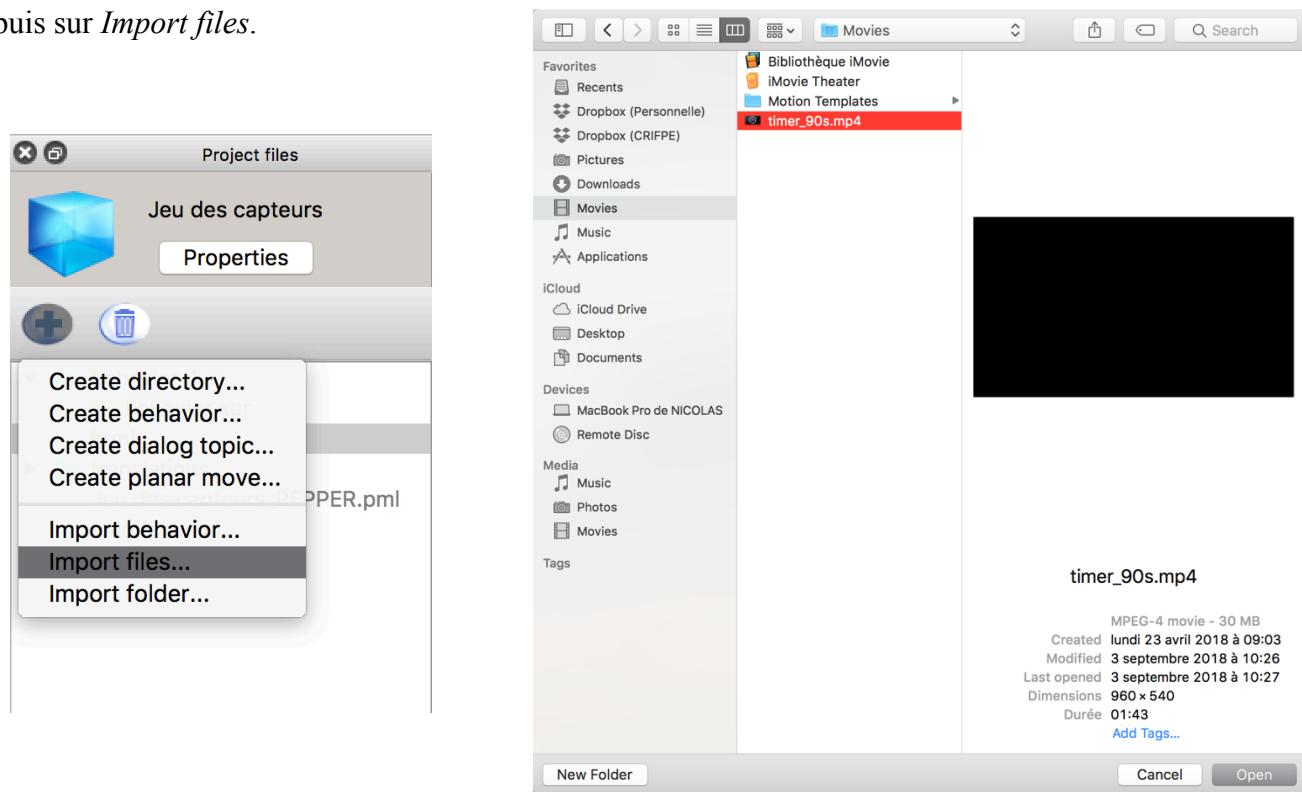
17. Maintenant, vous allez insérer une vidéo de compte à rebours au programme. Commencez par créer un dossier appelé *html*. Cliquez sur dans la fenêtre *Project files*, puis sur *Create directory*. Nommez le dossier *html* et appuyez sur *Create*.



Le dossier apparaît dans la fenêtre *Project files*.

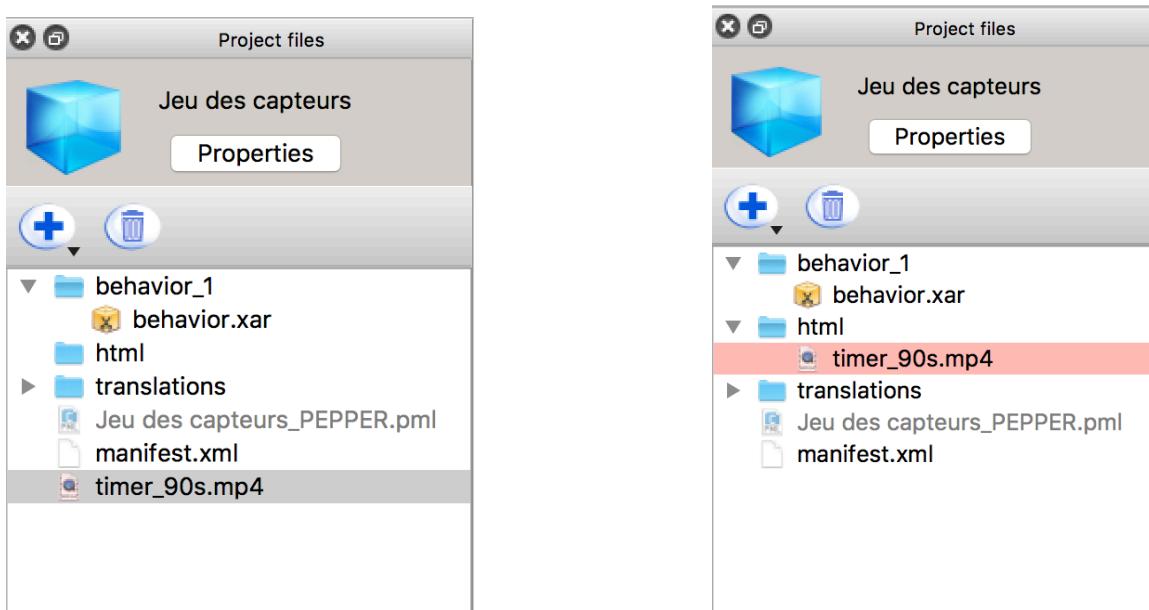


18. Vous allez ajouter la vidéo que vous souhaitez dans ce nouveau dossier. Cliquez sur  , puis sur *Import files*.



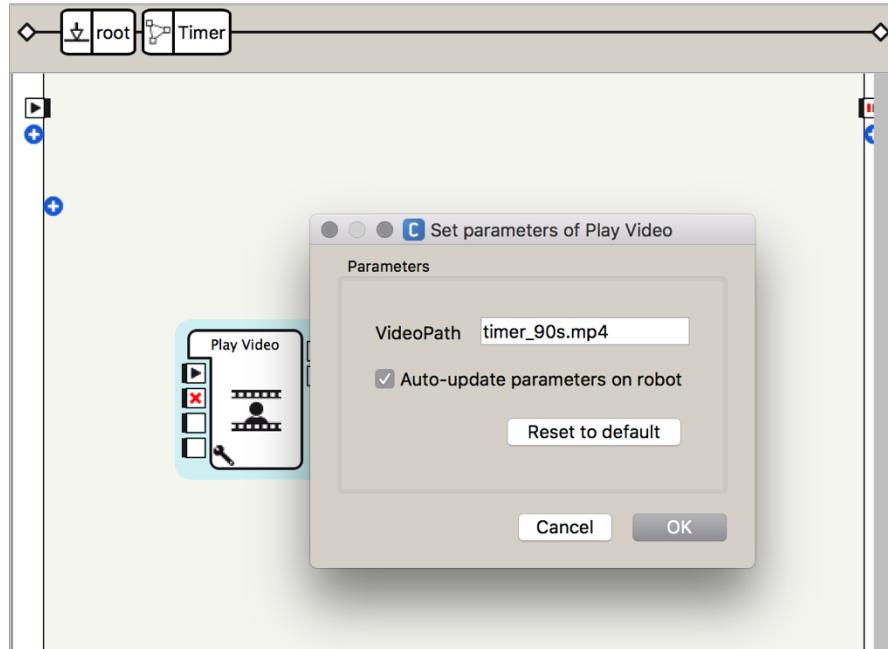
19. Sélectionnez la vidéo qui doit être au format *mp4* et appuyez sur *Open*. La vidéo apparaît dans la fenêtre *Project files* mais au mauvais endroit.

20. Placez le fichier dans le dossier *html*.

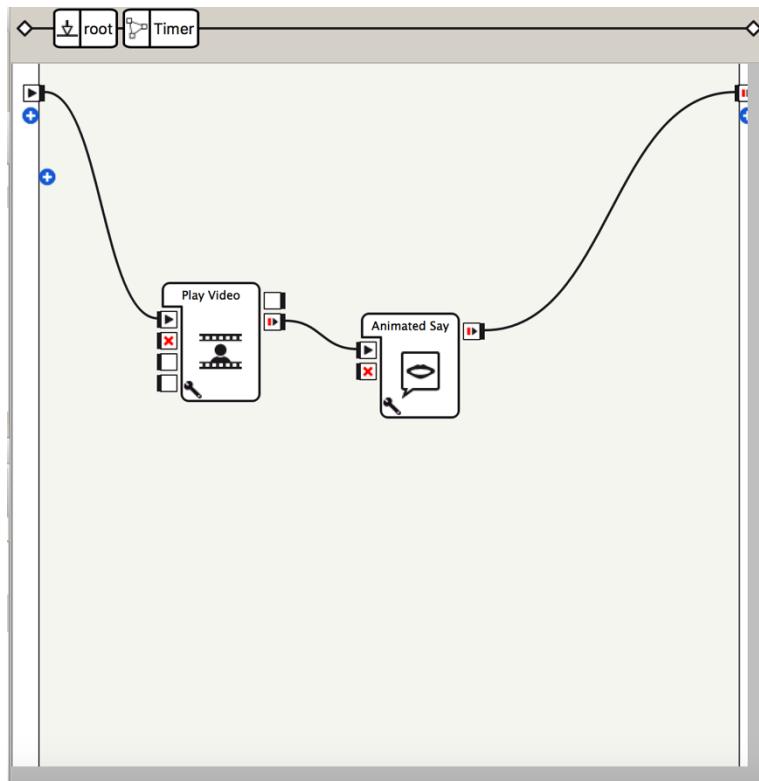


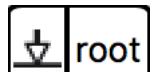


21. Cliquez sur la clé à molette de la boîte *Play Video* pour pouvoir ajouter le nom de la vidéo que vous avez ajoutée au projet.

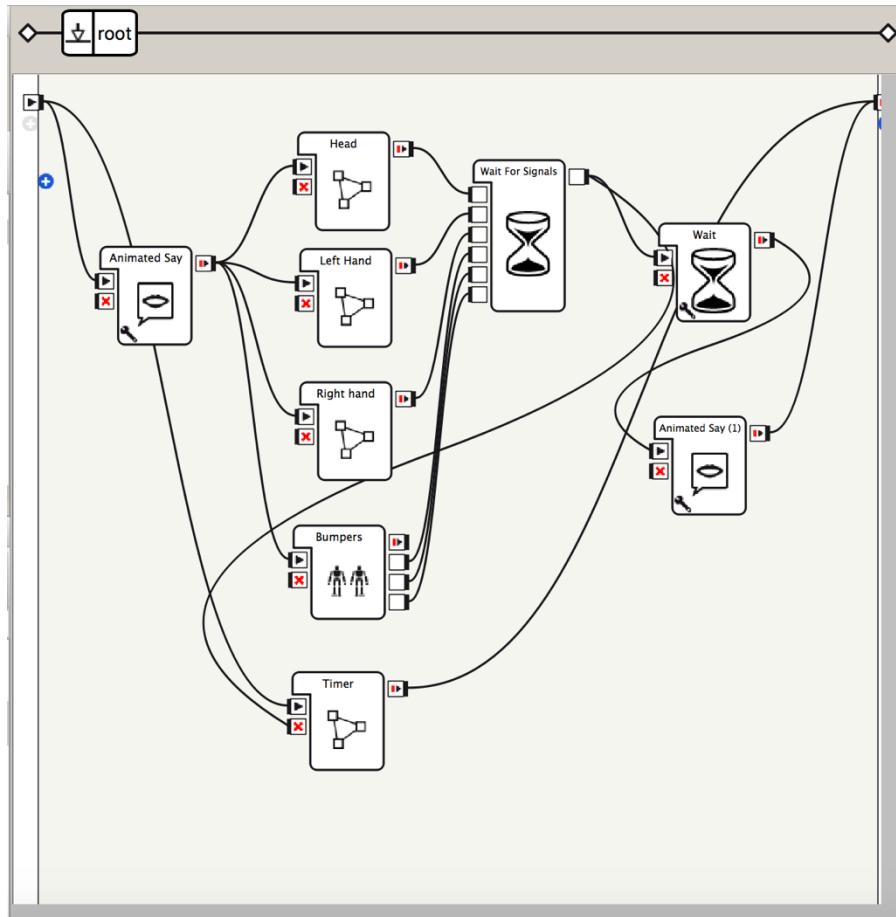


22. Insérer une boîte *Animated Say* dans la boîte *Diagram* afin de clôturer le jeu si jamais le compte à rebours s'écoule. Dans ce cas, le joueur perd la partie. N'oubliez pas de faire les connexions nécessaires.





23. Appuyez sur **root** pour revenir à l'espace de travail principal. Vous venez de terminer de programmer votre boîte *Timer*. Connectez là à l'*Onstart* et à l'*OnStopped*. Connectez également la sortie de *Wait for Signals* à l'entrée *OnStop* de votre boîte *Timer*. Cela permettra d'arrêter le compte à rebours si le joueur touche tous les capteurs.



24. Le programme est terminé, vous pouvez enregistrer le

programme dans le robot en appuyant sur . Le titre du programme apparaît dans la liste des applications et sur la tablette du robot.

