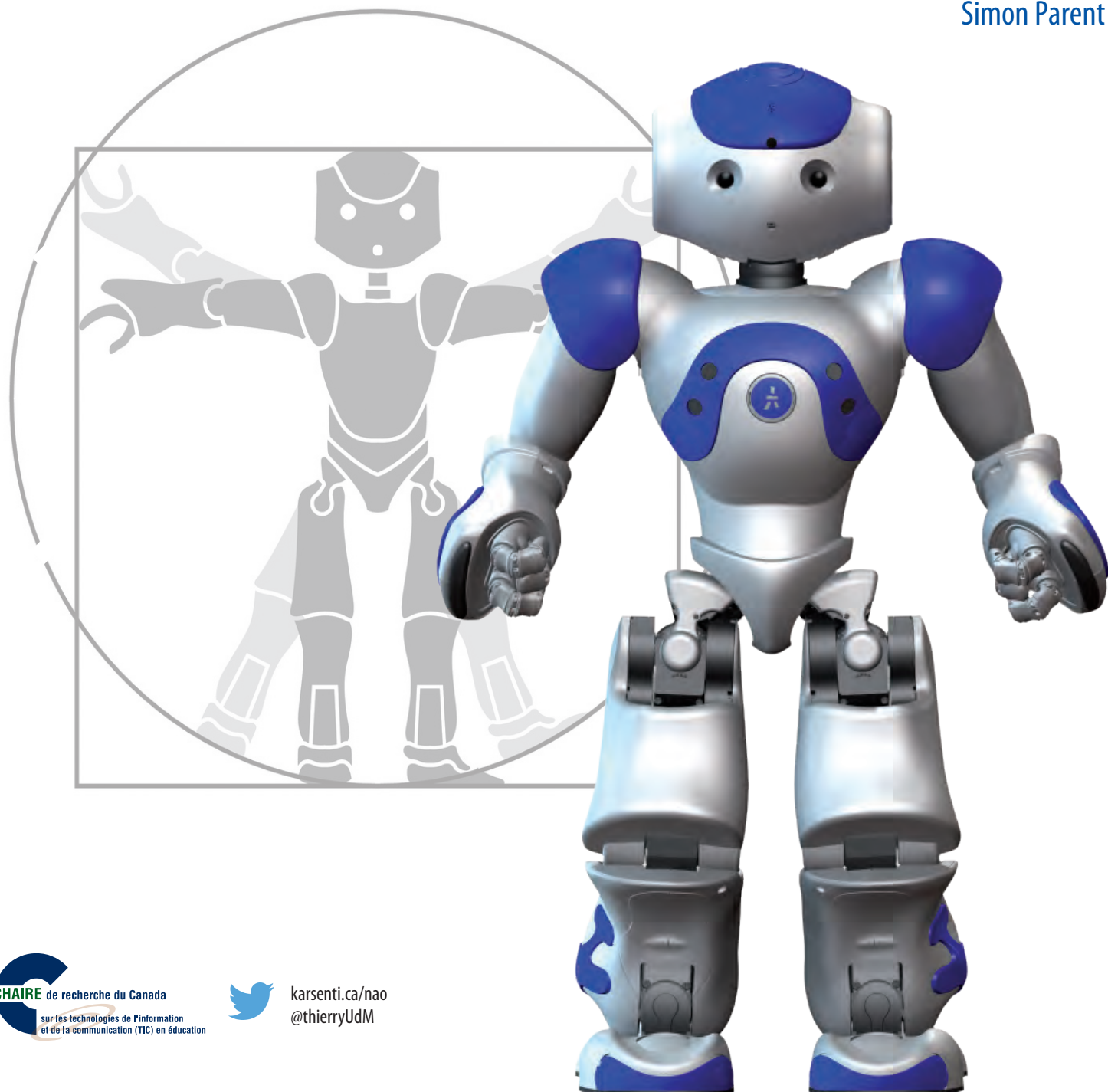


LE ROBOT NAO EN ÉDUCATION

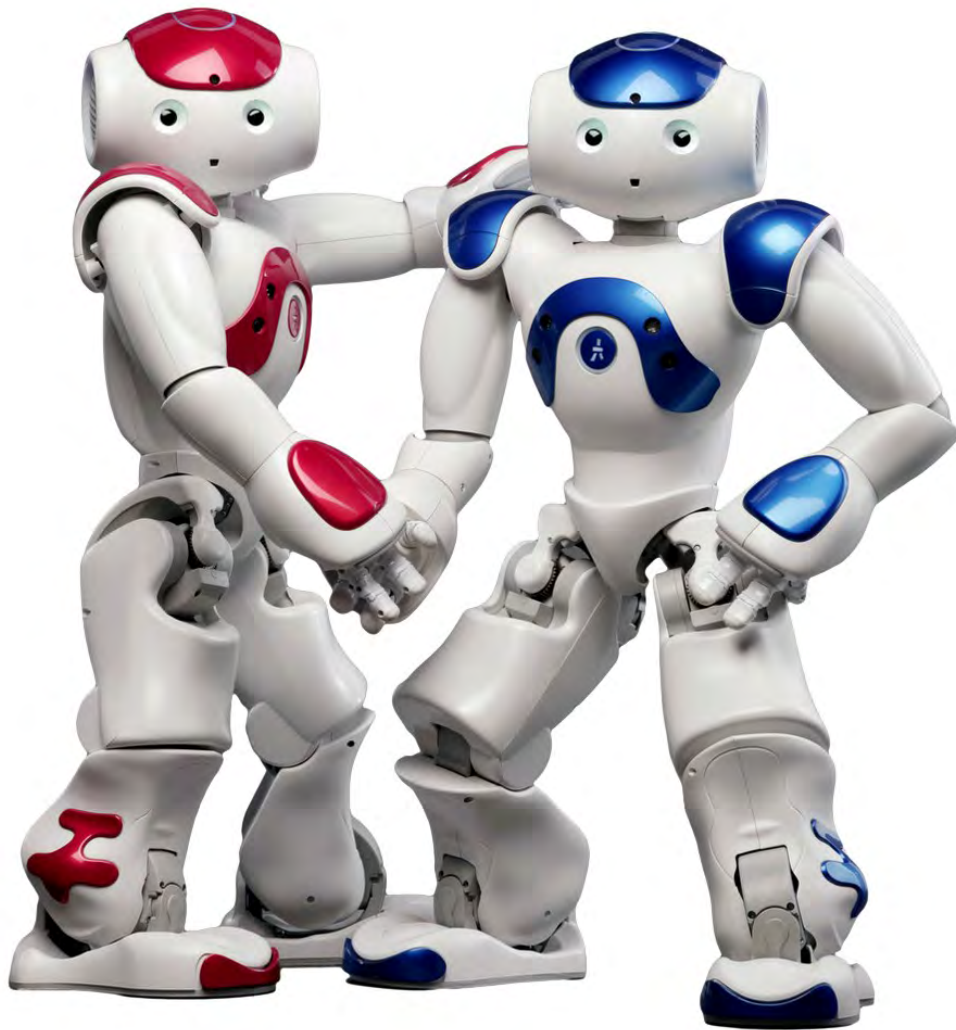
Guide de l'élève

Thierry Karsenti
Julien Bugmann
Simon Parent



LE ROBOT NAO EN ÉDUCATION

Guide de l'élève



Dépôt légal :

Bibliothèque et Archives Canada, 2017

ISBN : 978-2-923808-57-4

Ce document est publié sous une licence Creative Commons 4.0 de paternité (la moins restrictive).



Pour mieux comprendre ce type de licence, consultez le site creativecommons.ca

Pour citer ce document :

Karsenti, T., Bugmann, J. et Parent, S. (2017). *Le robot NAO en éducation. Guide de l'élève*. Montréal : CRIFPE.

Page couverture : Sylvie Côté.

Rapport disponible sur : karsenti.ca/nao/



L'usage du masculin n'est pas discriminatoire. Il a pour but d'alléger le texte.

TABLE DES MATIÈRES

PRÉFACE : NAO, LE ROBOT HUMANOÏDE	I
LES ROBOTS HUMANOÏDES EN ÉDUCATION	I
QUI EST NAO, LE ROBOT SOCIAL ET HUMANOÏDE?	I
Le fonctionnement du logiciel	1
La structure.....	1
Pour créer un comportement	6
Pour faire marcher NAO	11
Pour faire parler NAO	15
Pour faire jouer de la musique.....	18
Pour faire bouger ton robot NAO	22
En mode animation.....	22
Créer un mouvement.....	29
Comment créer un comportement	46
Créer une boîte	46
Comment utiliser la reconnaissance visuelle.....	54
Pour créer une base de données de reconnaissance.....	54
Pour utiliser la reconnaissance avec le robot.....	58
Pour faire reconnaître les visages.....	58
Comment utiliser la balle rouge	62
Comment utiliser la reconnaissance vocale	64
Questions et réponses avec NAO	64
Pour utiliser <i>Python</i>	67
Qu'est-ce que <i>Python</i> ?	67
La logique derrière le codage	67
Description du codage d'une boîte.....	68
Comment fonctionne le codage	70
Pour faire parler NAO avec <i>Python</i>	73
Lexique	76
<i>Audio</i>	76
<i>Behavior</i> (Comportement)	78
Communication	78
<i>Data edit</i> (Définir des données)	81
<i>Flow Control</i> (Contrôle de déroulement).....	82
<i>LEDs</i> (Diodes électroluminescentes).....	85
<i>MATH</i> (Mathématique).....	87
<i>Motions</i> (Mouvements)	89
<i>Sensing</i> (Détection).....	92
<i>System</i> (Système).....	95
<i>Templates</i> (Modèles)	96
<i>Trackers</i> (DéTECTEURS).....	97
<i>Vision</i>	99
<i>World representation</i> (Représentation du monde).....	102

PRÉFACE : NAO, LE ROBOT HUMANOÏDE

Depuis quelques années, de plus en plus de robots investissent les salles de classe, tant en Amérique, qu'en Asie ou en Europe. Qu'ils s'appellent *Bee-Bot*¹, *Dash*², *Mindstorm*³, ou encore *Sphero*⁴, plusieurs se questionnent sur leur réel potentiel éducatif auprès des élèves. En fait, plusieurs études ont montré que ces robots possèdent des caractéristiques qui font d'eux de puissants outils éducatifs, en particulier pour les élèves ayant des difficultés d'apprentissage. Parmi ces robots, il en est certains, comme les robots humanoïdes, qui semblent symboliser à eux seuls l'avenir du numérique en contexte scolaire.

LES ROBOTS HUMANOÏDES EN ÉDUCATION

Les robots humanoïdes ressemblent à un humain : ils ont un torse, une tête, deux bras et deux jambes. Certains ont même un visage qui ressemble (un peu) à celui d'un enfant, avec des yeux, une bouche, et une peau en silicone ou en plastique. De telles innovations technologiques commencent à être de plus en plus présentes dans la société et elles risquent de modeler davantage le monde de demain. Néanmoins, peu d'élèves de nos écoles du Québec ou d'ailleurs sont encore exposés à ces automates du futur.

QUI EST NAO, LE ROBOT SOCIAL ET HUMANOÏDE?

Mesurant 58 centimètres et pesant 4,8 kilogrammes, le robot social (il parle) et humanoïde (il répond spontanément à des questions et peut même reconnaître certains sentiments) NAO est composé de caméras, de capteurs, et de microphones, et il peut donc voir, entendre, reconnaître, et interagir avec des humains. Il n'est donc pas surprenant de voir plusieurs chercheurs amener le robot NAO à interagir avec des enfants⁵. Les travaux de certains chercheurs⁶ confirment d'ailleurs les facilités d'interaction rendues possibles par le robot NAO avec des enfants d'âge préscolaire et leur fort intérêt lorsque le robot se met en mouvement.

Ce guide vise à aider l'élève à gravir les niveaux proposés dans Maître NAO (page suivante). Pour toute question ou commentaire, n'hésitez pas à communiquer avec nous.

¹ <https://www.bee-bot.us>

² <https://www.makewonder.com/dash>

³ <https://education.lego.com>

⁴ <http://www.sphero.com/>

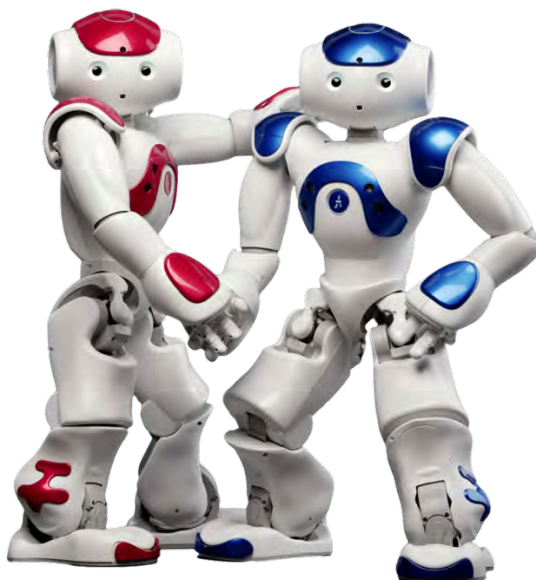
⁵ Shamsuddin, S., Yussof, H., Ismail, L., Hanapiah, F. A., Mohamed, S., Pia, H. A. et Zahari, N. I. (2012). Initial response of autistic children in human-robot interaction therapy with humanoid robot NAO. Dans *2012 IEEE 8th International Colloquium on Signal Processing and its Applications (CSPA)* (p. 188-193). IEEE.

⁶ Ioannou, A., Andreou, E. et Christofi, M. (2015). Pre-schoolers' interest and caring behaviour around a humanoid robot. *TechTrends*, 59(2), 23-26.

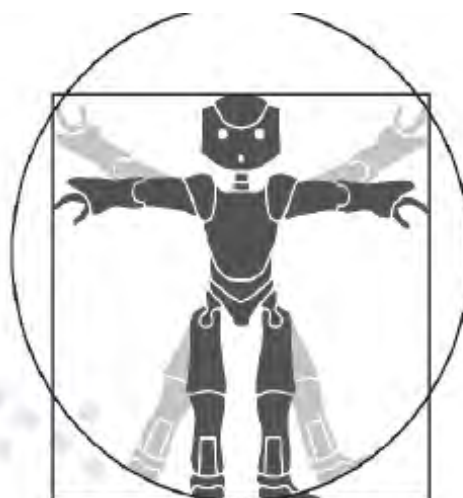
Devenez le maître NAO



Niveaux	Défis	Aide	Niveaux	Défis	
01 Jaune	<ul style="list-style-type: none"> Vous devrez doucement caresser le dessus de la tête de NAO pour qu'il réagisse Vous devrez dire « Bonjour » ou « Salut » à NAO jusqu'à ce qu'il vous comprenne et vous réponde Vous devrez demander à NAO : « Comment vas-tu ? » ou « Comment tu t'appelles » jusqu'à ce qu'il vous comprenne et vous réponde 	Interne au robot	06 Rouge	<ul style="list-style-type: none"> Vous devrez faire lever le bras gauche de NAO. Vous devrez faire lever les deux bras de NAO. Vous devrez donner une nouvelle position à NAO sans qu'il ne tombe. 	Guide p.22
02 Orange	<ul style="list-style-type: none"> Vous devrez faire dire « Bonjour » à Nao en le programmant. Vous devrez faire dire un « Bonjour » animé (salut de la main) à NAO en le programmant. Vous devrez faire s'asseoir NAO en le programmant. 	Guide p.15	07 Bronze	<ul style="list-style-type: none"> Vous devrez faire en sorte que NAO reconnaisse une balle rouge. Vous devrez faire en sorte que NAO reconnaisse une balle rouge, se dirige vers elle et s'arrête à 0,3 m. Vous devrez faire en sorte que NAO reconnaisse une balle rouge, se dirige vers elle, s'arrête et dise la phrase : « Ma balle est ici, je la cherchais justement ». 	Guide p.62
03 Vert	<ul style="list-style-type: none"> Vous devrez lui faire tourner la tête à gauche. Vous devrez lui faire lever le bras droit. Vous devrez lui faire bouger les deux bras en même temps. 	Guide p.37	08 Argent	<ul style="list-style-type: none"> Vous devrez faire apprendre 2 visages (ou plus) à NAO. Vous devrez faire en sorte que NAO reconnaisse votre visage et vous appelle par votre prénom. Vous devrez faire en sorte que NAO reconnaisse votre visage et vous dise un message personnalisé (différent pour chaque personne). 	Guide p.58
04 Bleu	<p>UTILISER LA BOÎTE MOVE TO POUR CE NIVEAU</p> <ul style="list-style-type: none"> Vous devrez faire avancer NAO d'un mètre. Vous devrez faire reculer NAO d'un mètre. Vous devrez faire avancer NAO de 0,5 mètre, lui faire dire bonjour en effectuant un salut de la main et lui faire dire « Content de vous rencontrer ». 	Guide p.11	09 Or	<p>UTILISER LA BOÎTE DE TYPE PYTHON SCRIPT POUR CE NIVEAU</p> <p>Vous devrez créer une boîte de type Python Script et la déposer dans l'espace de travail. Vous devrez ensuite remplacer les informations de la boîte (nom, description, image) :</p> <ul style="list-style-type: none"> Le nom de la boîte devient « Bonjour robotici » ; La description devient : « NAO fait un salut spécial » ; L'image devient box script.png, dans le dossier d'images de Chronographe. <p>Vous devrez faire dire à NAO : « Bonjour jeune homme, je suis content de vous voir après tout ce temps passé dans ma boîte » en utilisant la boîte de type Python que vous avez créée à la première étape.</p> <p>Vous devrez :</p> <ul style="list-style-type: none"> Saisir dire quelque chose à NAO (de votre choix) renommer votre boîte, y ajouter une description et une image personnalisée lui faire avancer de 0,5m lui faire faire un mouvement particulier 	Guide p.73
05 Violet	<ul style="list-style-type: none"> Vous devrez faire poser la question suivante à NAO : « Aimes-tu les desserts ? » Vous devrez faire poser la question suivante à NAO : « Quelle est la capitale du Canada ? ». Vous devrez proposer 3 choix de réponse et vous devrez faire en sorte que NAO reconnaisse et félicite la bonne réponse et demande de réessayer en cas de mauvaise réponse. Vous devrez faire poser la question suivante à NAO : « Quelle est la capitale des Etats-Unis ? » Proposez 3 choix de réponse et NAO devra lever le bras droit en cas de bonne réponse et lever le bras gauche en cas de mauvaise réponse. 	Guide p.65	10 Platinum	<ul style="list-style-type: none"> Vous devrez faire jouer une musique de votre choix à NAO (*fichier mp3 seulement). Vous devrez lui faire jouer cette musique et lui faire faire une chorégraphie avec les bras (Mode Timeline). Vous devrez lui faire jouer une musique et lui faire faire une chorégraphie avec les bras et les jambes. (Mode Timeline). 	Guide p.18 et 22



Attestation de passage de niveau **MAÎTRE NAO**



Niveaux

01	Jaune ◆◆◆◆	◆	◆◆◆	◆◆◆◆◆
02	Orange ◆◆◆◆			
03	Vert ◆◆◆◆			
04	Bleu ◆◆◆◆			
05	Violet ◆◆◆◆			
06	Rouge ◆◆◆◆			
07	Bronze ◆◆◆◆			
08	Argent ◆◆◆◆			
09	Or ◆◆◆◆			
10	Platinum ◆◆◆◆			



LE LOGICIEL CHOREGRAPHE

PARTIE 1

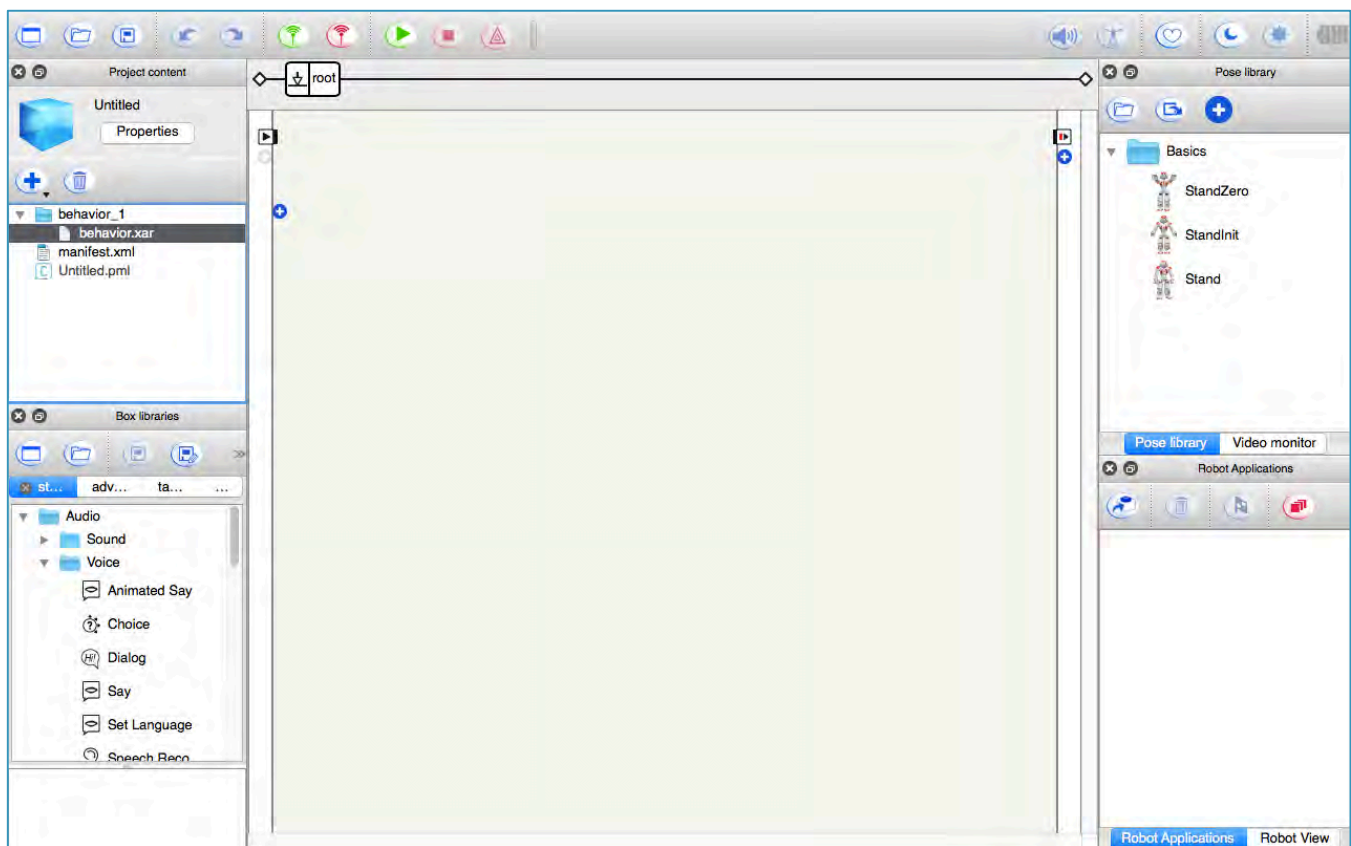
LE FONCTIONNEMENT DU LOGICIEL

SOURCE : LE ROBOT NAO - [HTTP://STETEL2.FREE.FR/AN2016/AFF_RUB.PHP](http://steteil2.free.fr/AN2016/AFF_RUB.PHP)



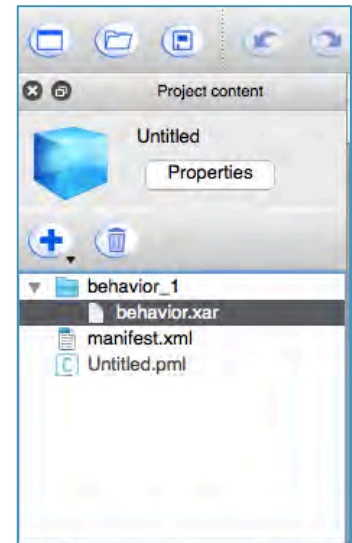
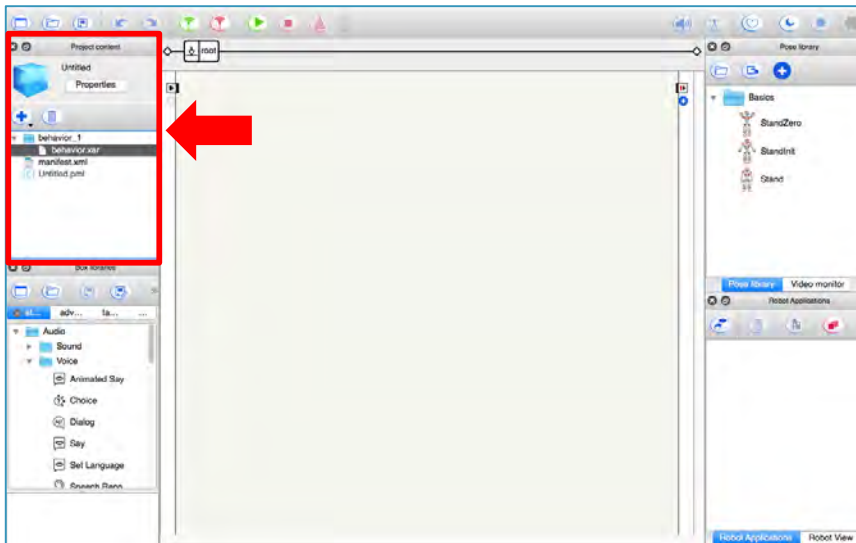
LA STRUCTURE

Voici l'interface du logiciel *Choregraphe*.

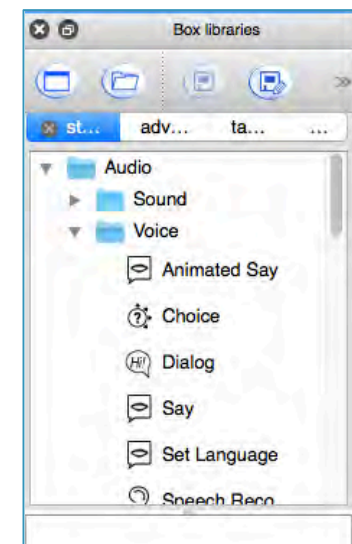
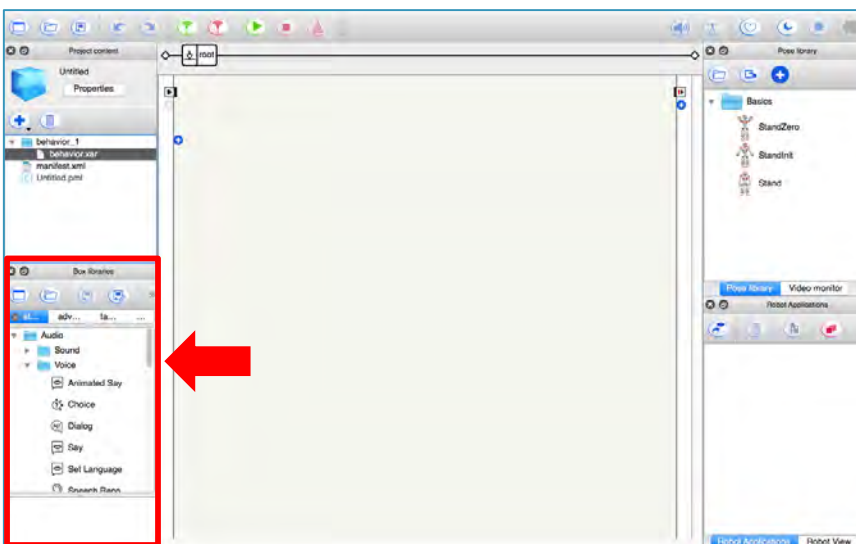


Dans la **colonne de gauche**, tu peux trouver :

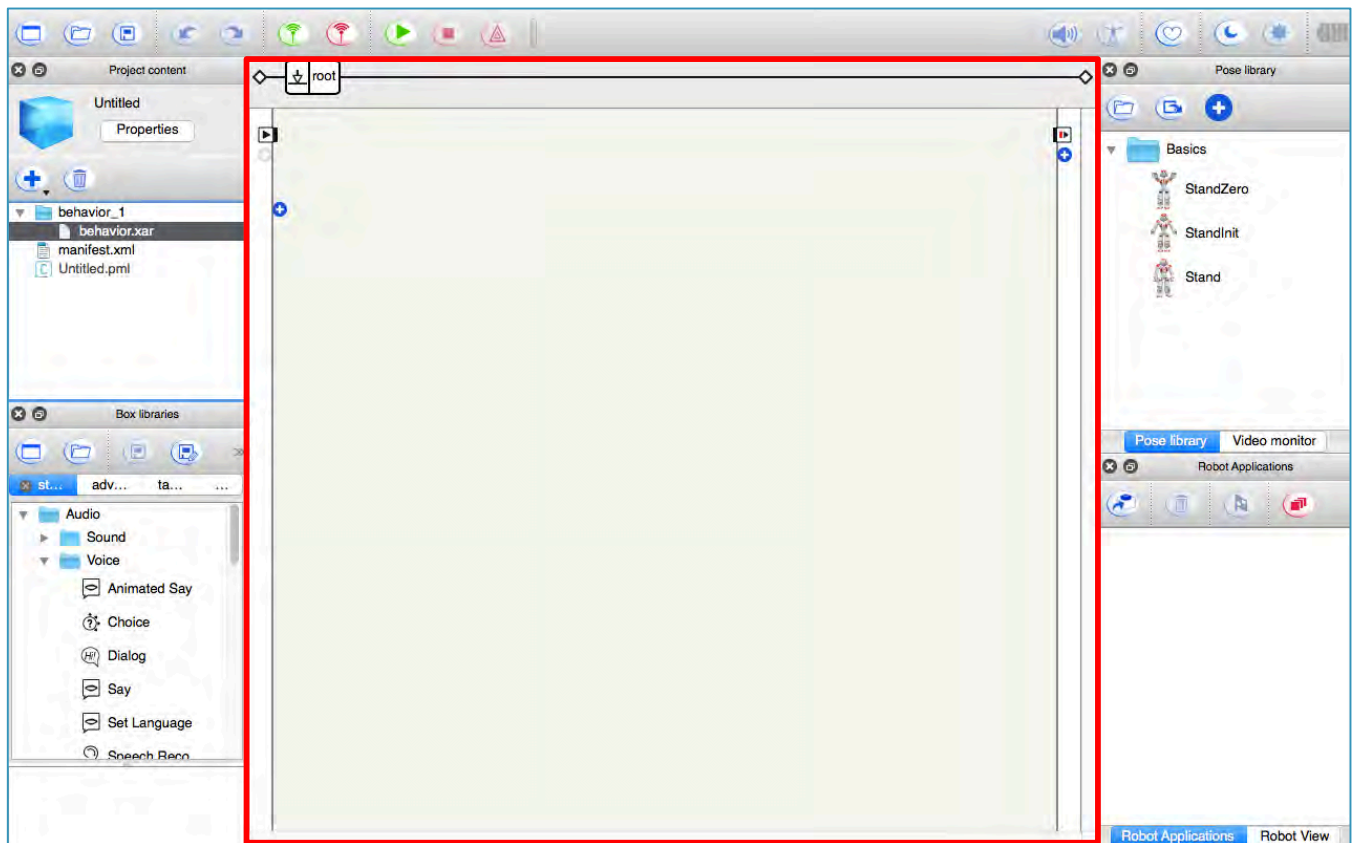
1. La fenêtre *Project content* : ce sont les éléments qui se retrouvent dans ton projet.



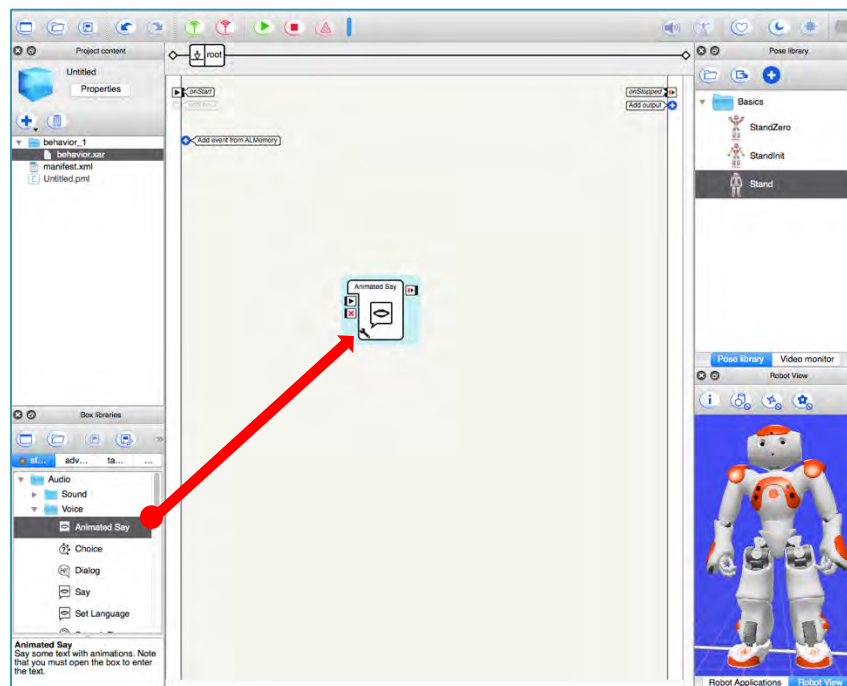
2. La fenêtre *Box libraries* : tu pourras y trouver toutes les **actions (box)** que tu peux ajouter dans ton espace de codage.



La fenêtre au centre est ton espace de travail (l'endroit où tu vas faire ton codage).

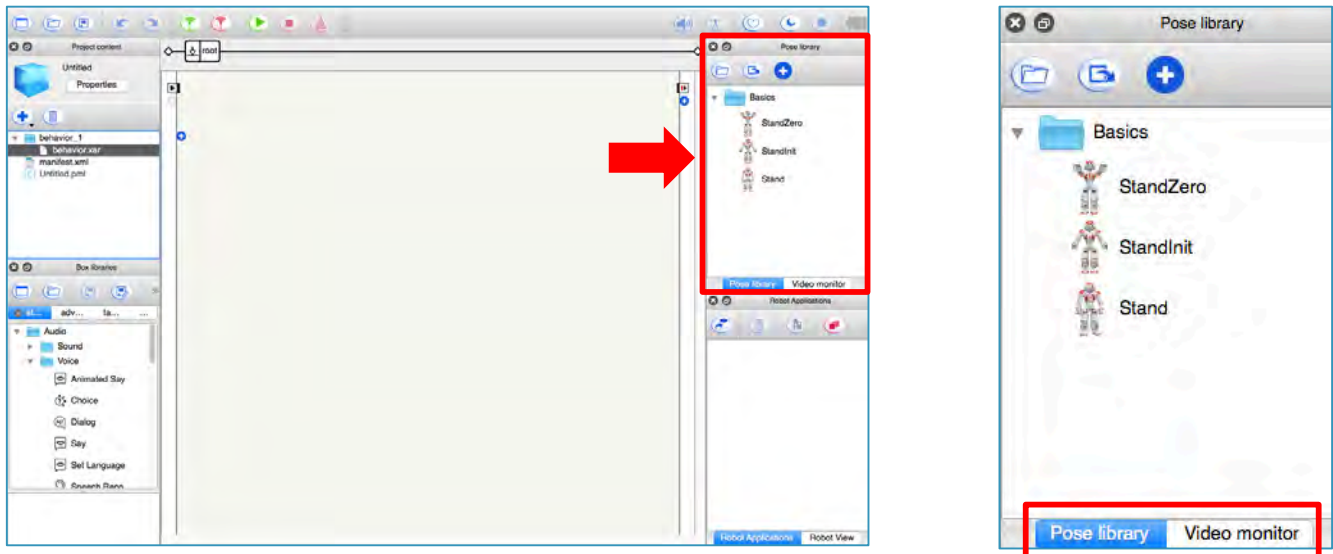


Tu peux sélectionner des boîtes (box) dans les colonnes de gauche et de droite, puis **les déposer dans l'espace de travail**. Par exemple :

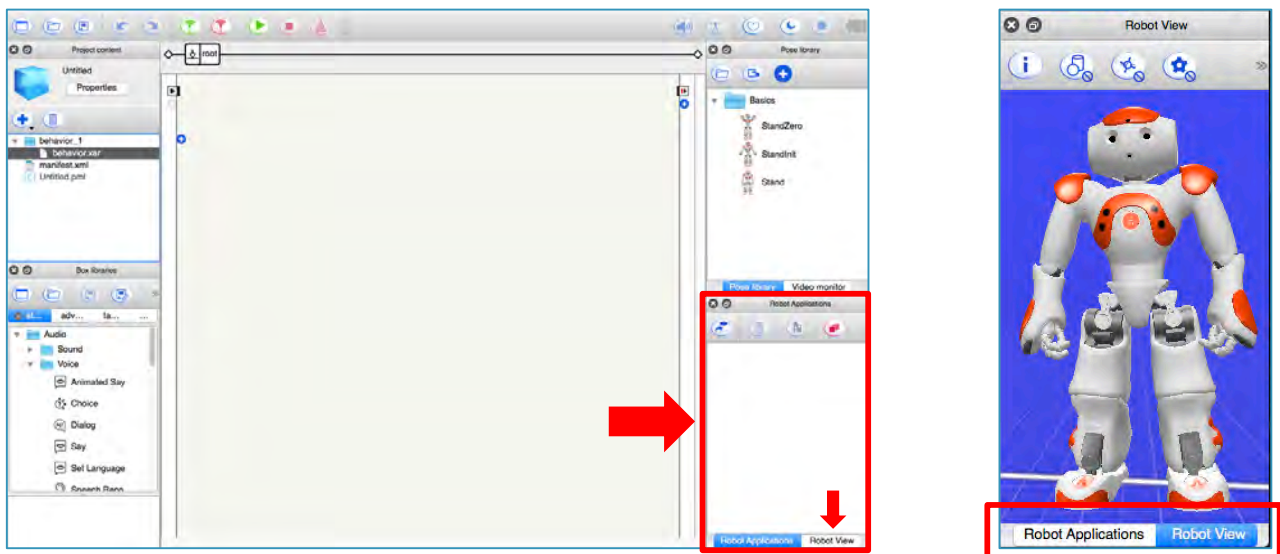


Dans la **colonne de droite**, tu peux trouver :

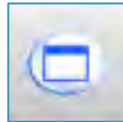
1. La fenêtre *Pose library* : tu peux trouver plusieurs **poses que le robot peut faire**. Si tu cliques sur l'onglet *Vidéo monitor*, tu auras accès à la caméra du robot.



2. La fenêtre *Robot view* : en cliquant sur l'onglet *Robot view*, tu peux **voir ton robot** faire les mouvements que tu as programmés dans ton espace de travail (simulation).



La barre d'outils :



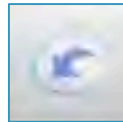
Créer un nouveau projet



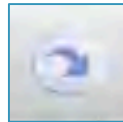
Ouvrir un projet



Enregistrer le projet



Annuler l'action que tu viens de faire



Rétablir l'action



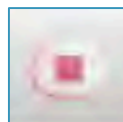
Connecter un robot



Déconnecter un robot



Jouer une séquence



Arrêter une séquence

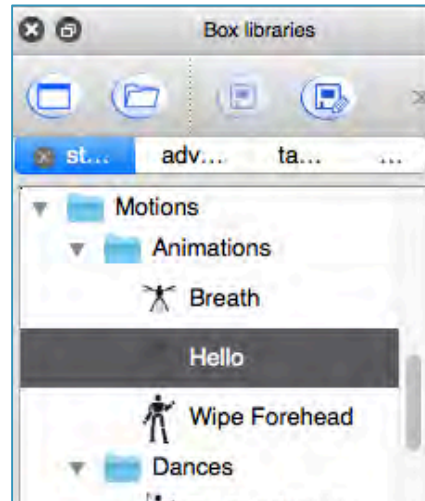


Erreurs

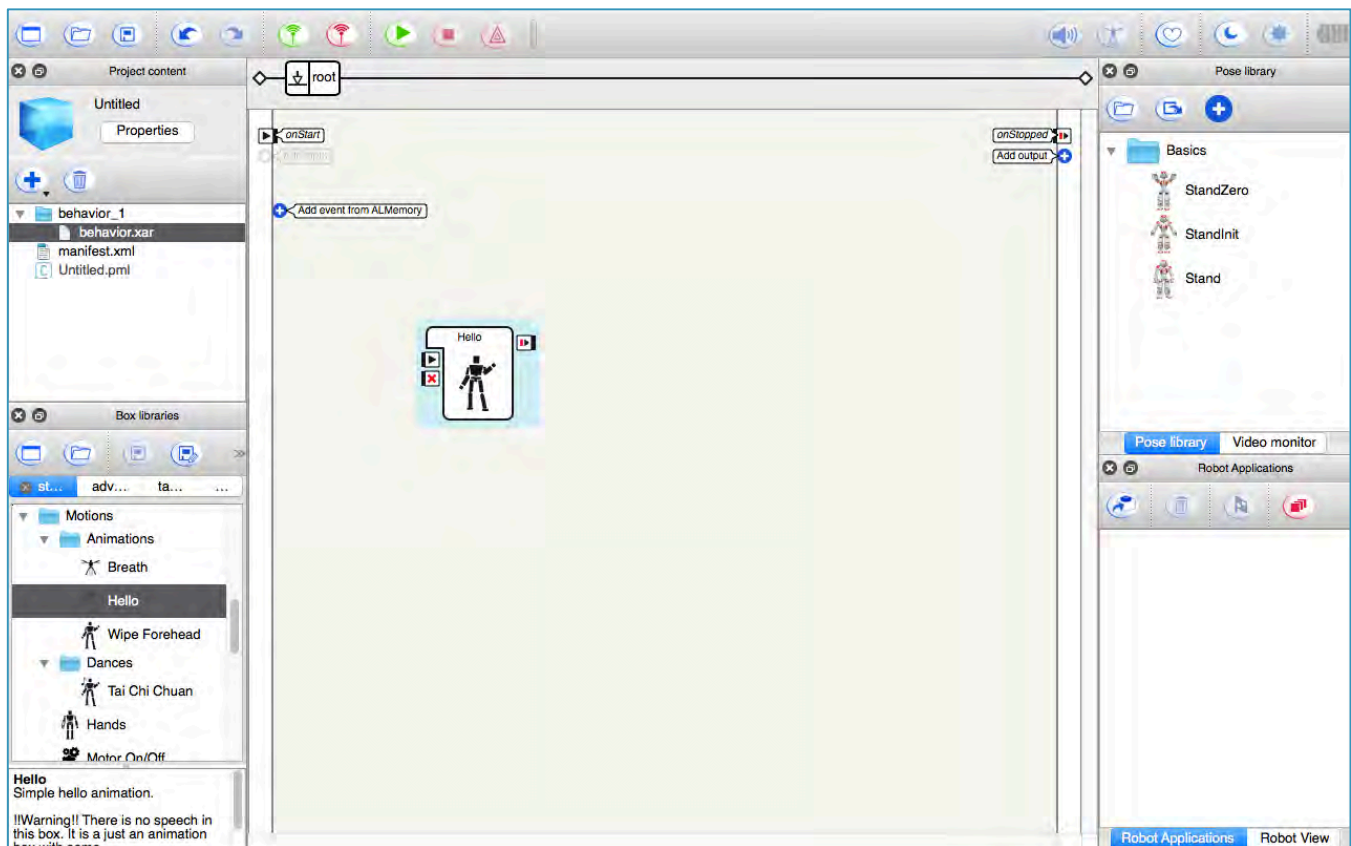
POUR CRÉER UN COMPORTEMENT



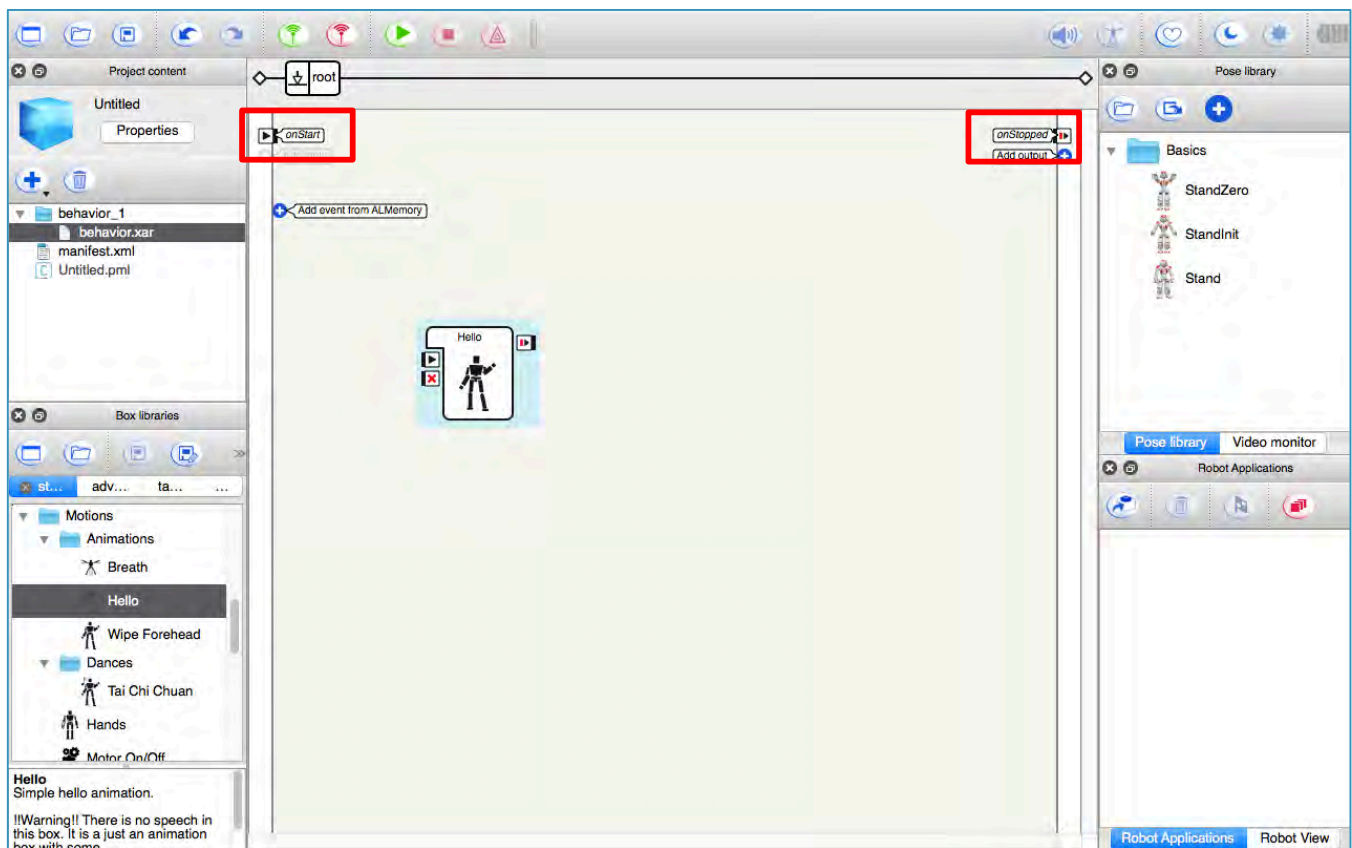
1. Dans la fenêtre *Box libraries*, cherche le dossier MOTIONS.
2. Ouvre ensuite le sous-dossier ANIMATIONS.



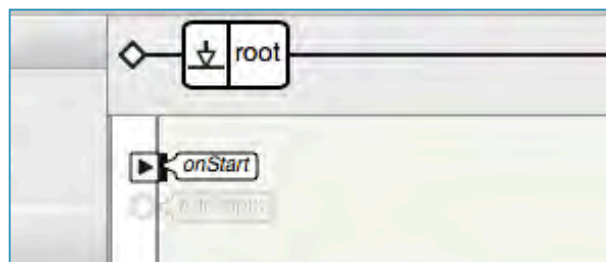
3. Sélectionne la boîte *HELLO* et dépose-la dans ton espace de travail.



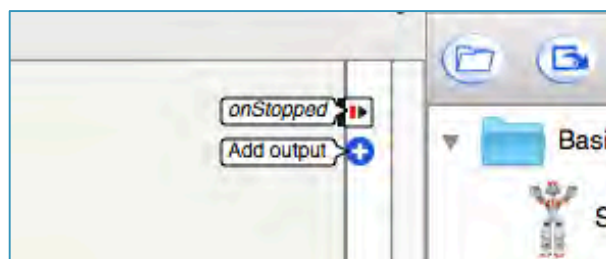
Pour faire bouger ton robot, il faut que tu connectes toutes les actions ensemble. On va appeler l'ensemble des actions « une **séquence** ».



Dans ton espace de travail, tu vois le **début de la séquence** (*onStart*) à gauche :



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

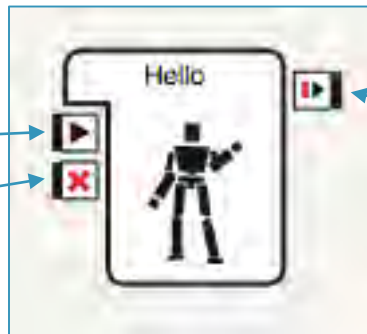


Chaque boîte contient plusieurs connexions :

Les entrées (début)

Start

Stop

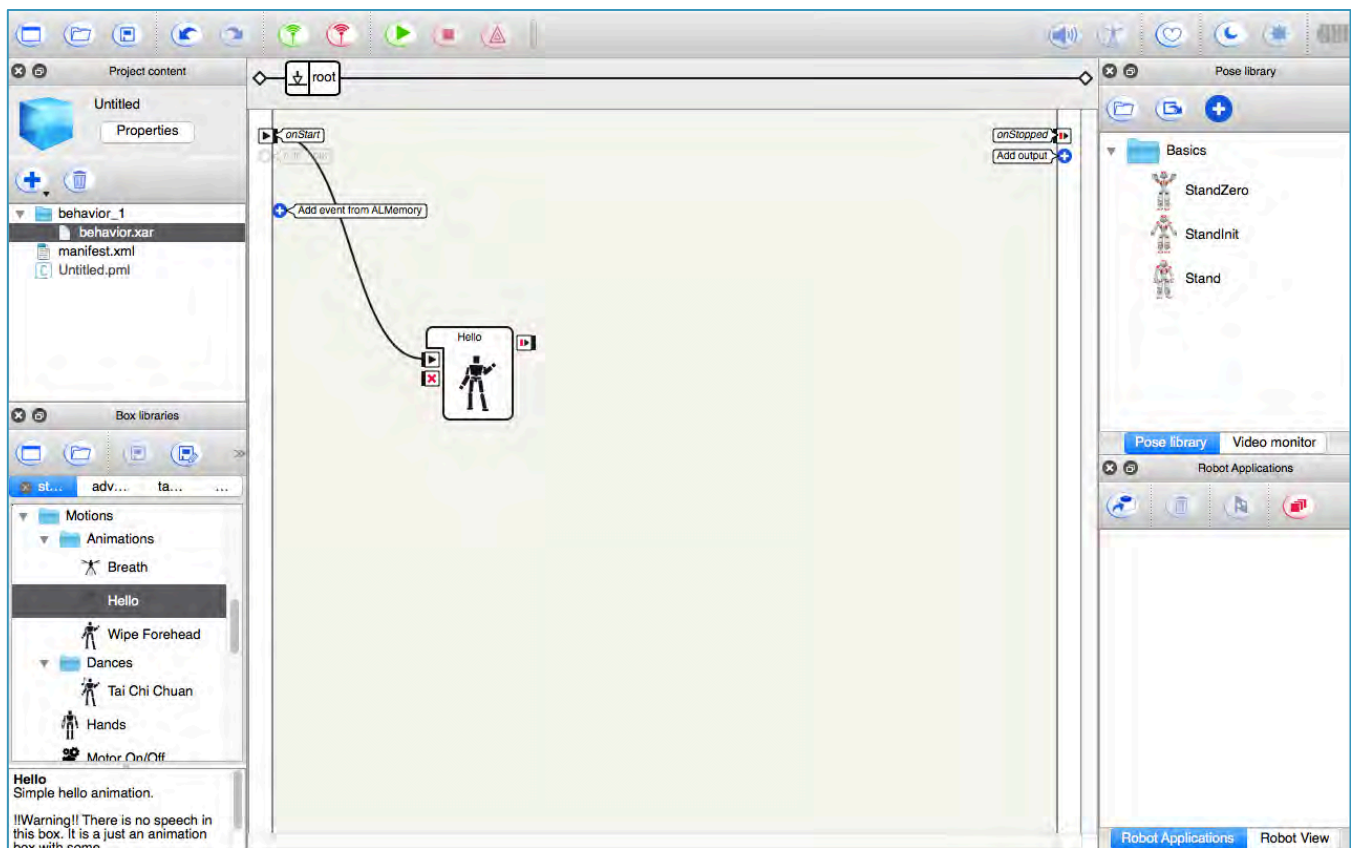


La sortie (fin)

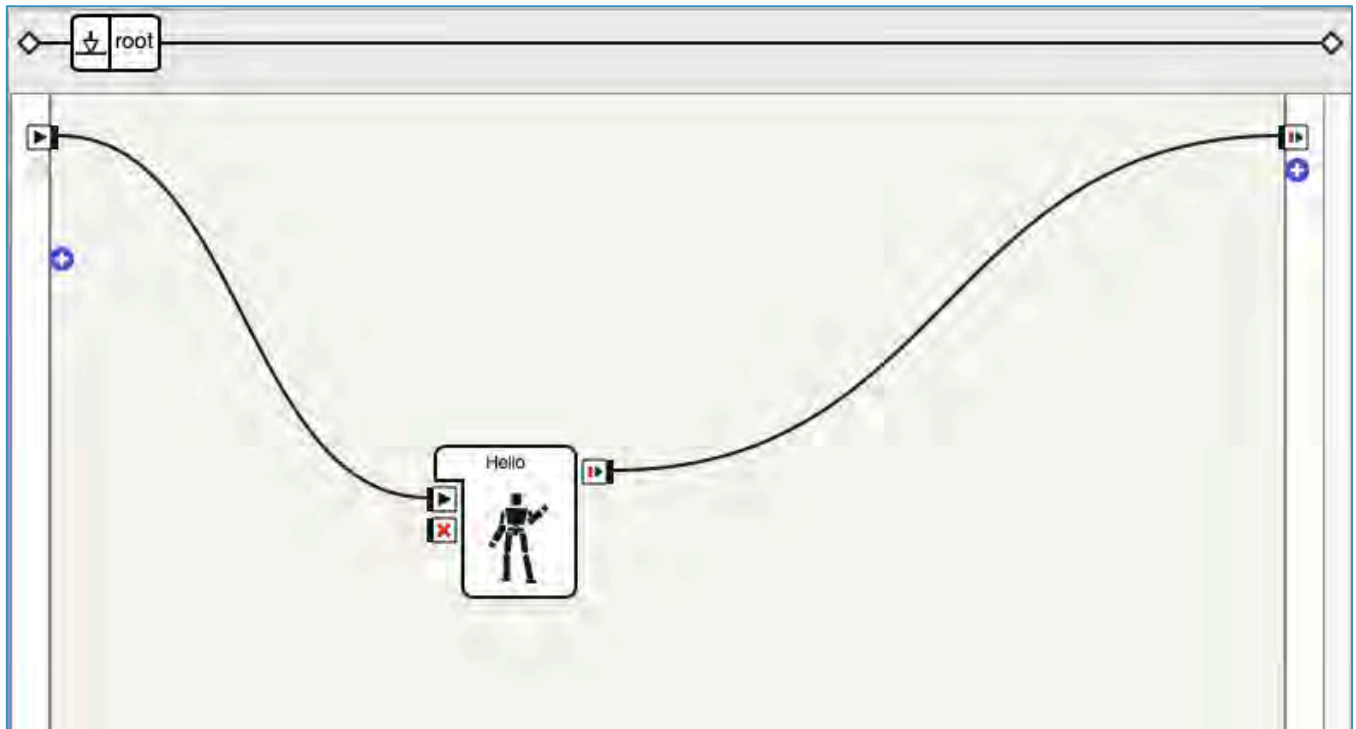
Fin de l'action



4. Appuie sur le début (*onStart*) et rends-toi jusqu'à la boîte. Connecte le lien à la connexion *START*.



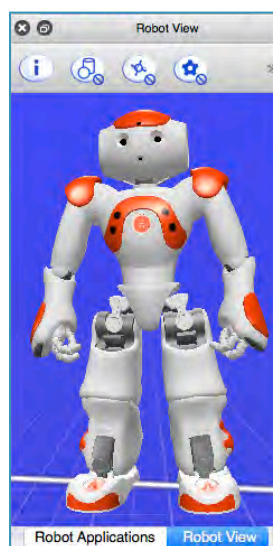
5. Maintenant, relie ta boîte à la fin (*onStopped*).



6. Pour vérifier que ta séquence fonctionne, appuie sur le bouton

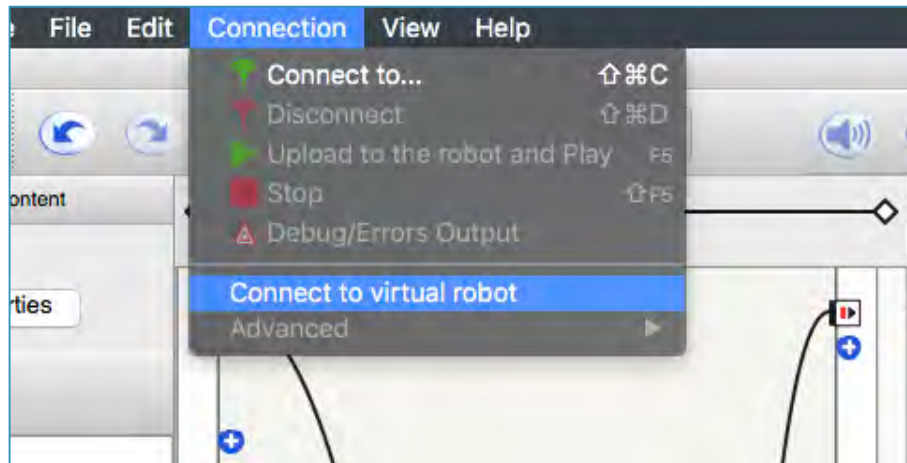


Tu peux regarder la **simulation des mouvements** du robot dans la fenêtre *Robot view*.

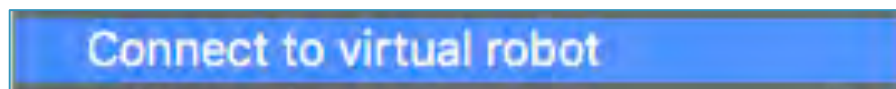




Si rien ne s'affiche dans la fenêtre *Robot View*...



Dans la barre de menu, appuie sur *Connection*, puis sur :



Ton robot NAO virtuel (simulation) va apparaître dans la fenêtre *Robot View* :





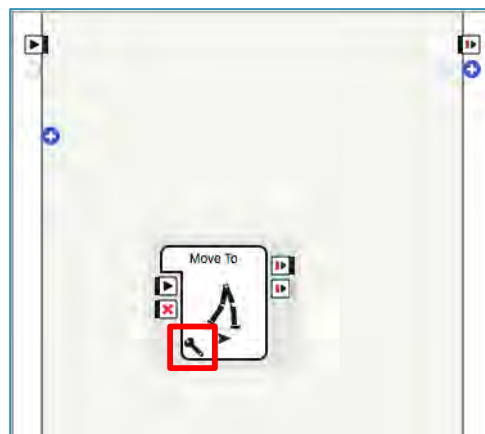
POUR FAIRE MARCHER NAO

SOURCE : ALDEBARAN DOCUMENTATION, CHOREGRAPHE TUTORIALS

Dans la fenêtre *Box libraries*, sélectionne le dossier *Motions*, puis, repère l'action *Move To* :



Appuie sur *Move To* et déplace la boîte dans **ton espace de travail** (fenêtre du centre).

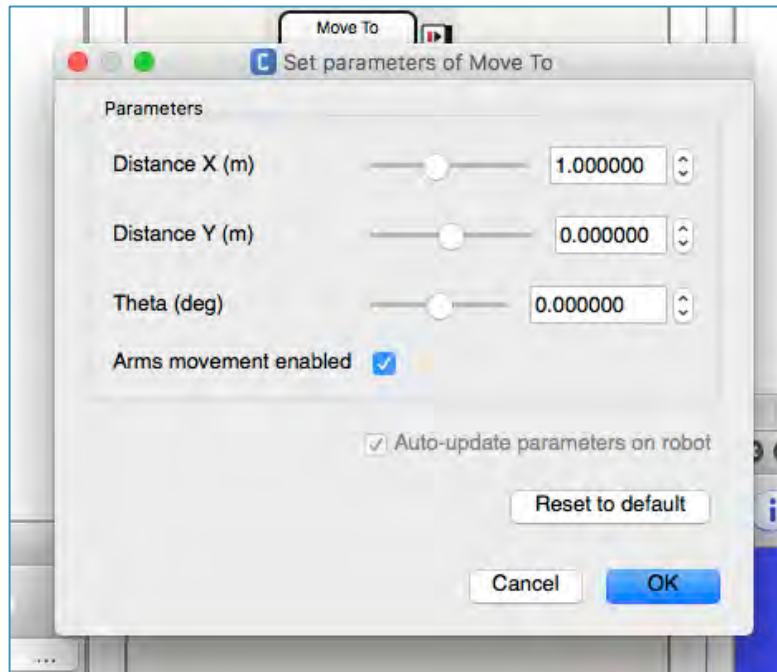


Appuie ensuite sur



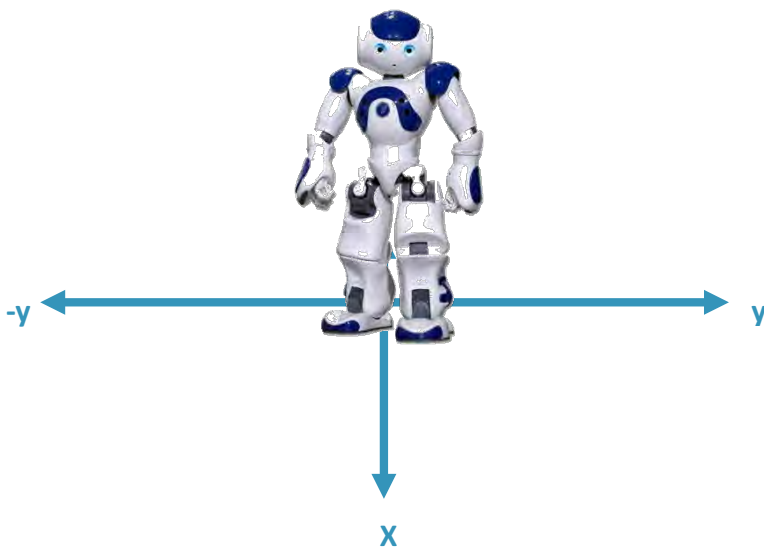
pour modifier les paramètres de l'action.

Une nouvelle fenêtre va apparaître.

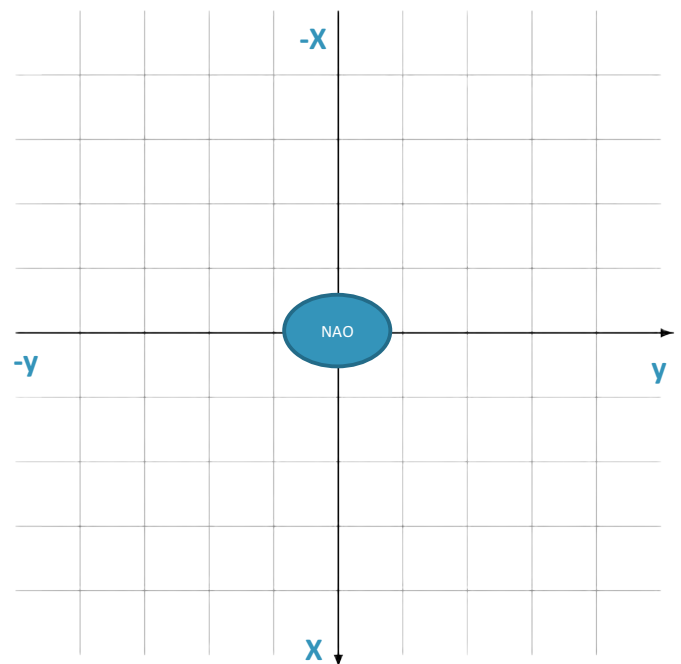


Imagine ton robot NAO au centre de deux axes (X et Y).

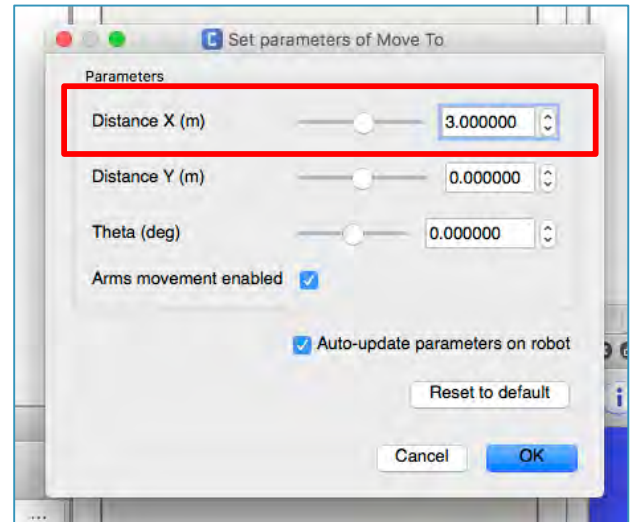
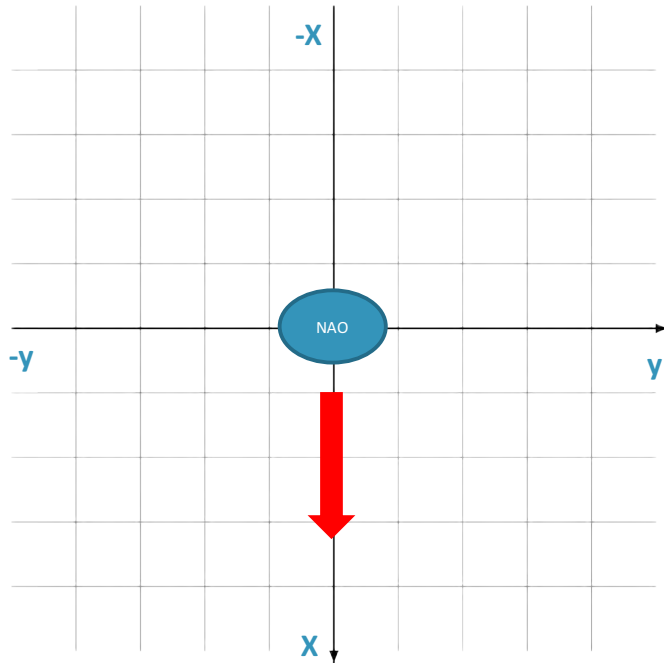
Vue de face



Vue de haut

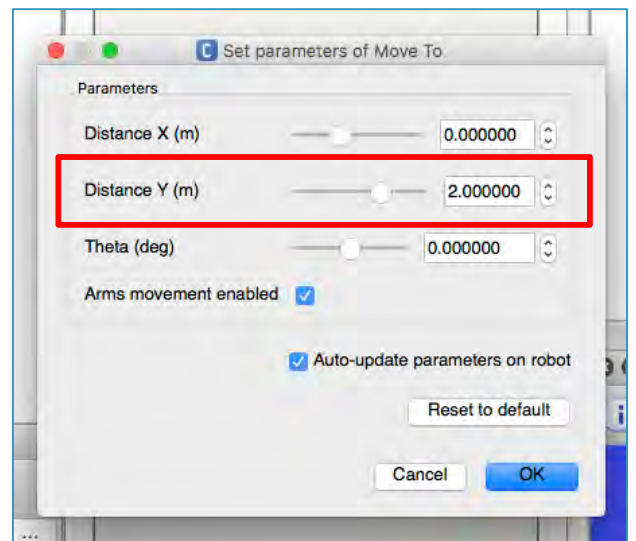
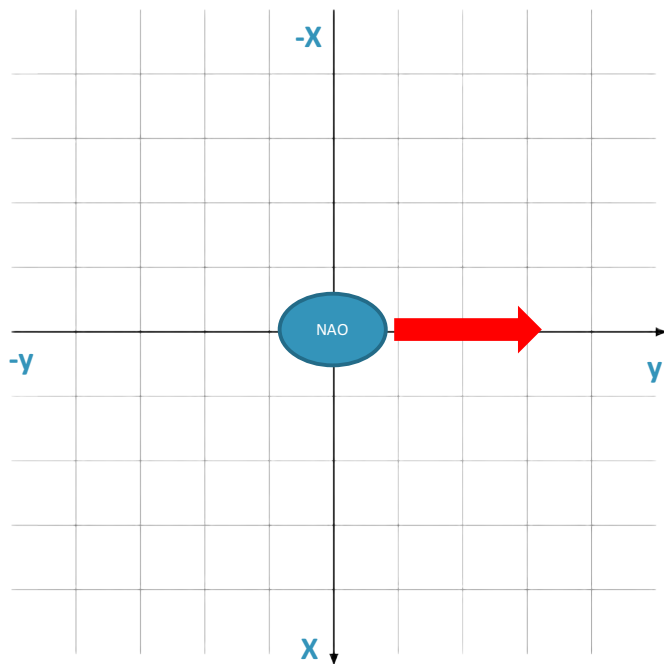


Pour **faire avancer** ton robot en avant, **augmente la valeur de X** ($X > 0$ et $Y = 0$):



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

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

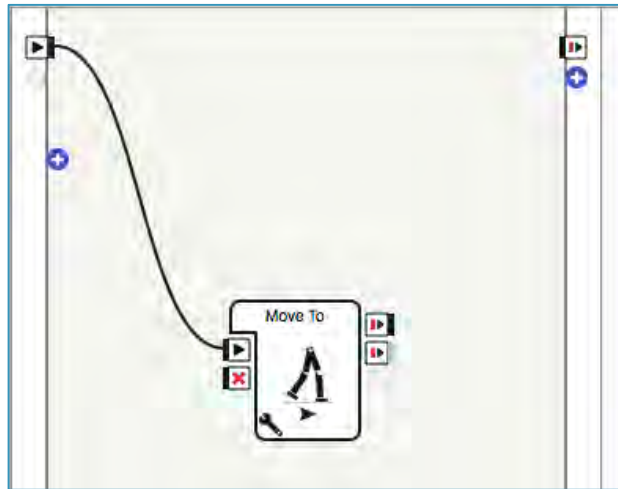


Dans l'exemple, mon robot va se déplacer à gauche de 2 mètres, puisque $Y = 2,000000$.

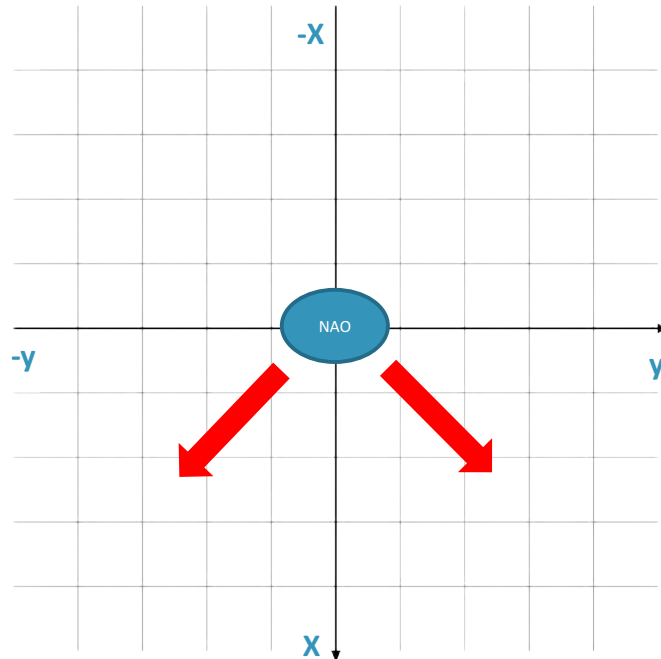


ATTENTION

1. N'oublie pas de faire ton lien avec onStart



2. Si la valeur de X **et** de Y ont une valeur autre que 0 **en même temps**, ton robot avancera en diagonale.

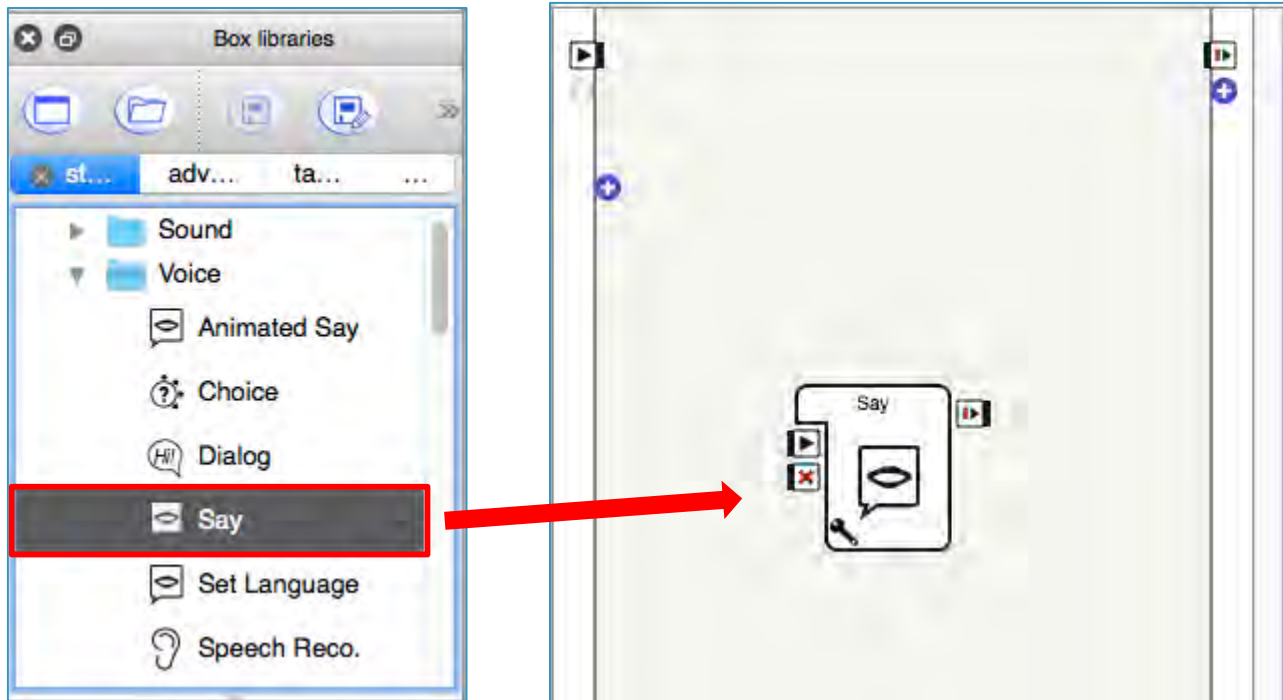


3. Pour le **faire reculer** ou pour le **déplacer vers la droite**, tu dois utiliser les valeurs négatives de X et Y.



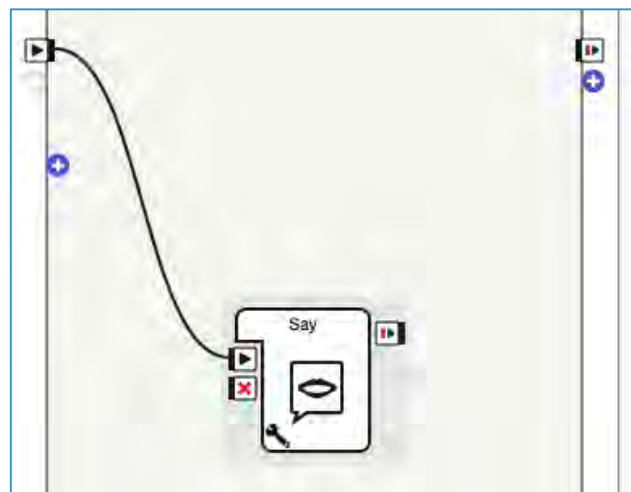
POUR FAIRE PARLER NAO

Dans la fenêtre *Box Libraries*, cherche le dossier *Audio*, puis *Voice* :



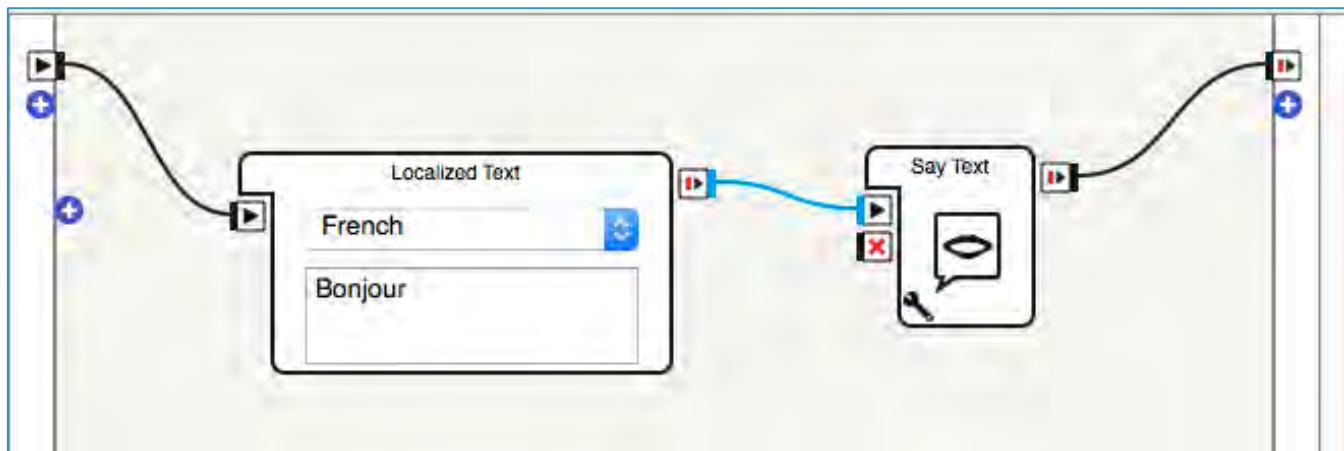
Déplace ensuite la boîte Say dans ton espace de travail.

Tu peux ensuite **connecter ta boîte** avec le début de ta séquence :

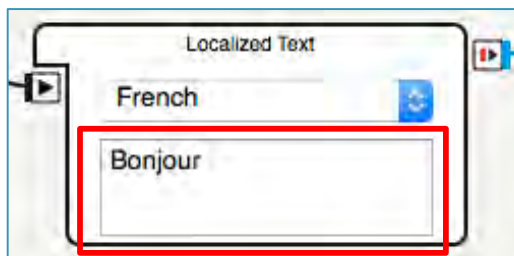


Double-clique sur la boîte **Say** dans ton espace de travail.

Tu vois apparaître les boîtes suivantes :

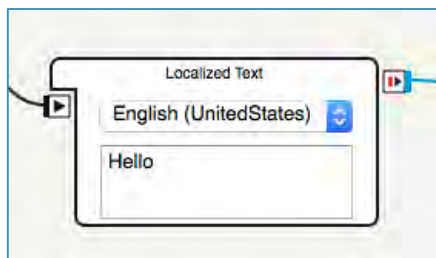


Dans la boîte *Localized Text*, écris le texte que tu veux faire dire à ton robot NAO.

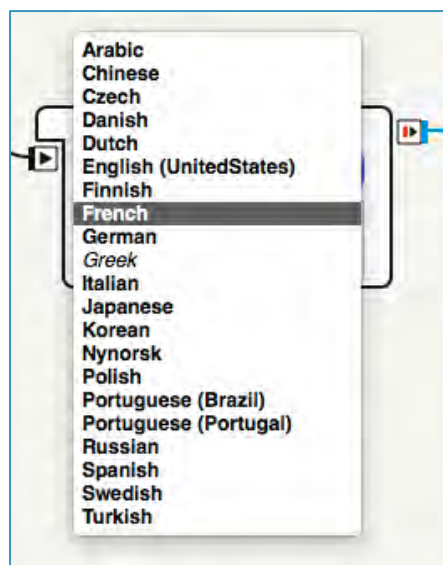


ATTENTION

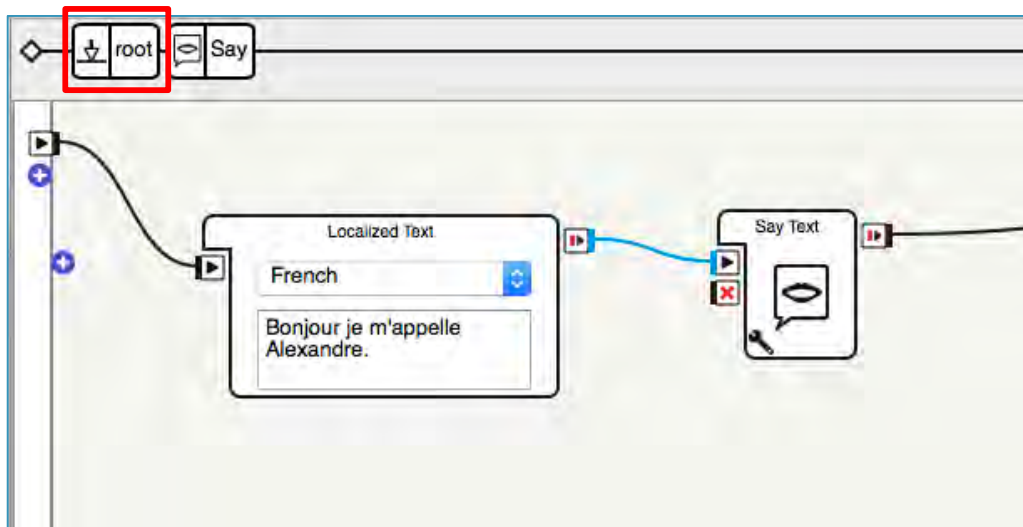
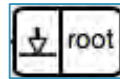
Il est possible que la langue par défaut soit l'anglais :



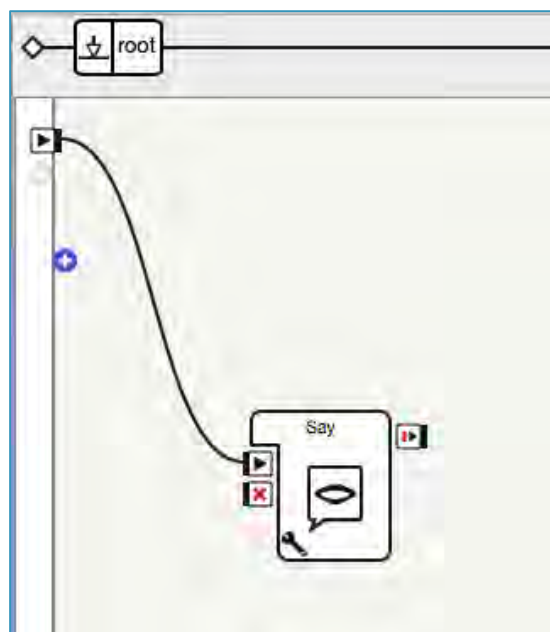
Tu peux la changer en cliquant sur le menu déroulant :



Quand tu as **terminé d'écrire ton texte**, appuie sur



Tu vas revenir à ton **espace de travail principal**, où apparaîtra ta séquence complète :



Pour **tester ta séquence**, appuie sur

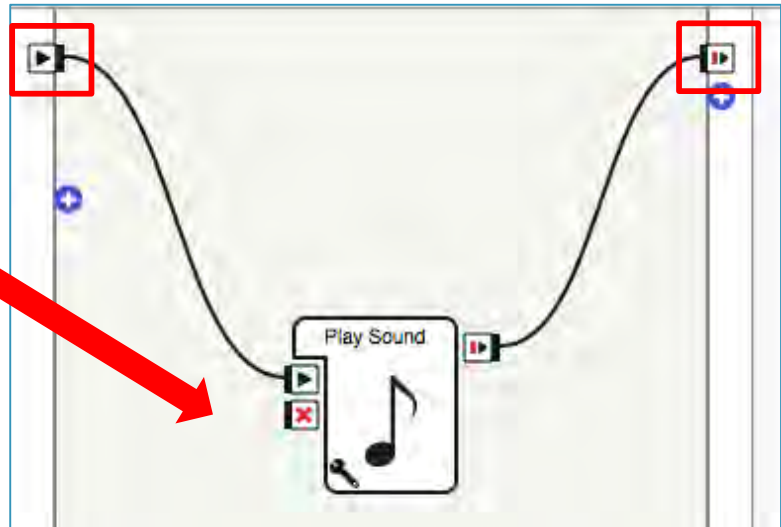
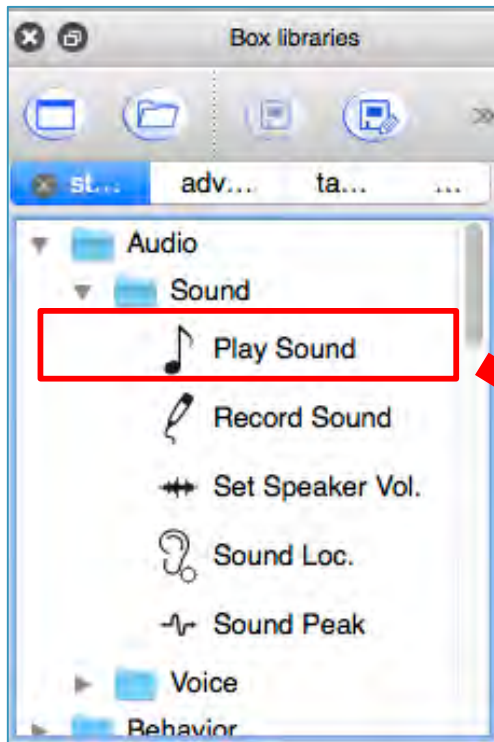


(Attention : Tu dois tester la séquence **avec le vrai robot** pour entendre la voix)




POUR FAIRE JOUER DE LA MUSIQUE

Dans la fenêtre *Box Libraries*, cherche le dossier *Audio*, puis *Sound*.



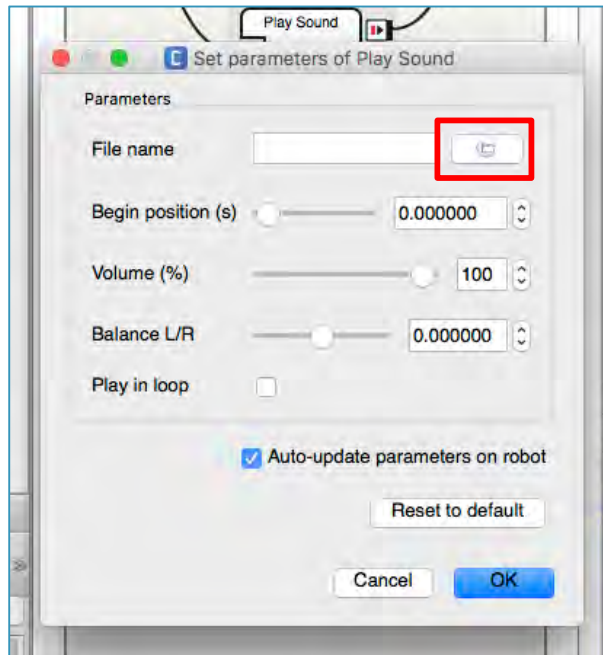
Sélectionne *Play Sound* et **déplace la boîte dans ton espace de travail**.

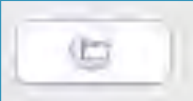
* N'oublie pas de connecter ta boîte à *onStart* et à *onStop*.

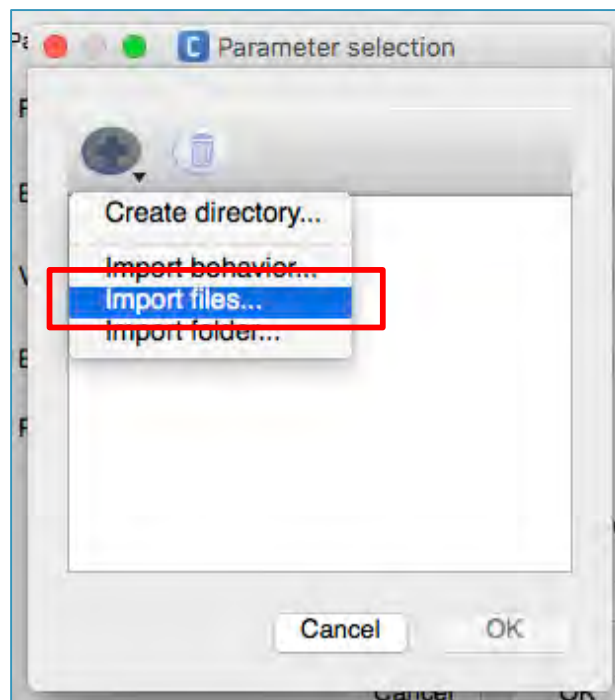
Appuie sur  pour **modifier les paramètres** :




Une fenêtre va apparaître :

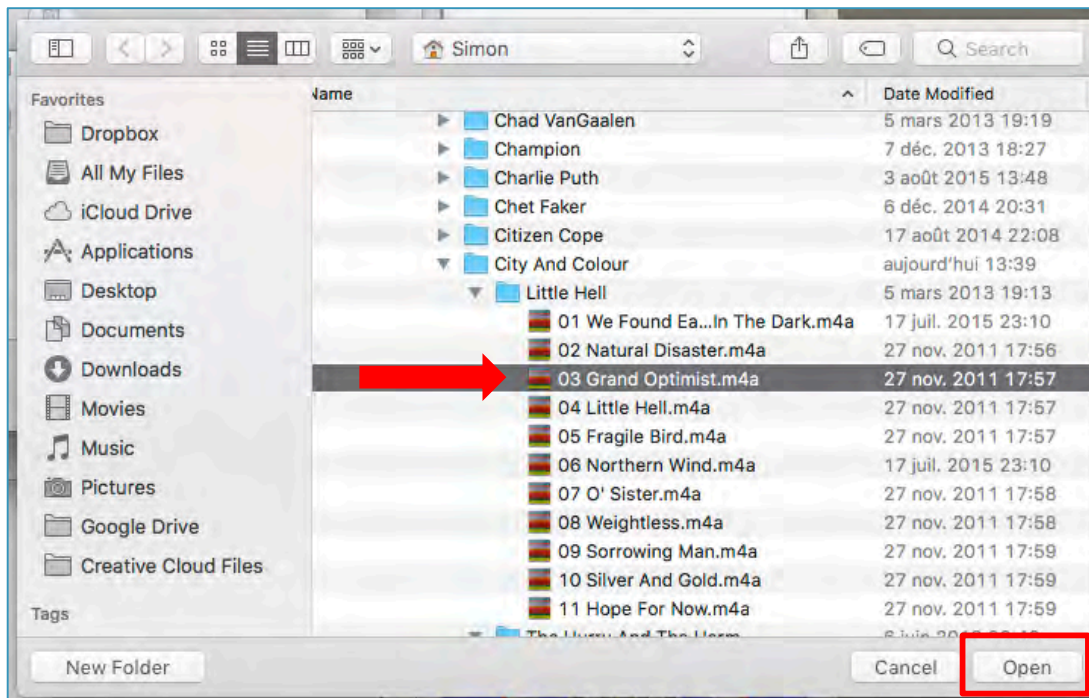


Appuie sur  pour chercher un fichier (une musique).

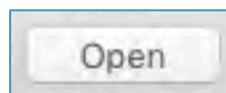


Appuie sur  et choisis l'option **Import files...**

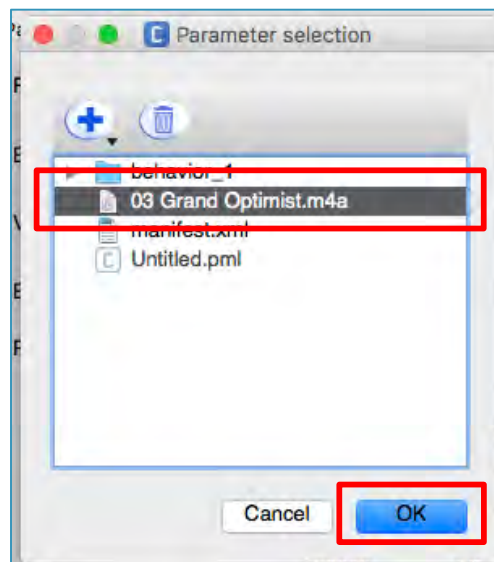
Tu peux maintenant **choisir un fichier** dans ton ordinateur.



Sélectionne le fichier et appuie sur

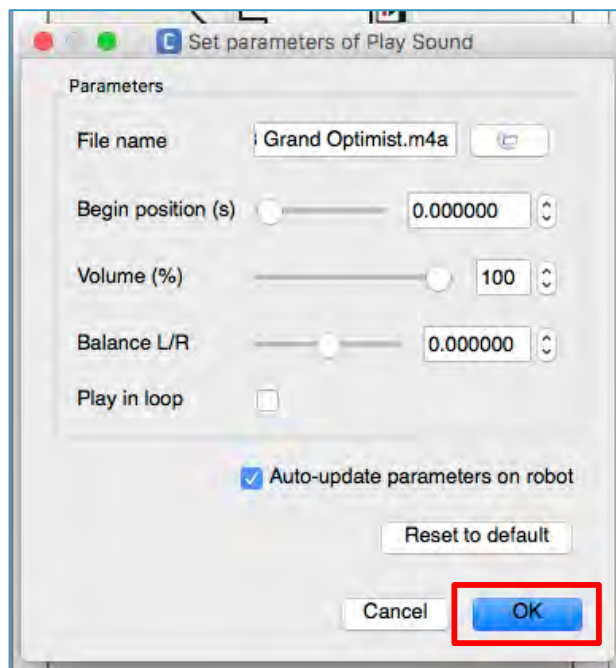


Ton fichier apparaît maintenant dans ton projet :



Assure-toi qu'il soit **sélectionné** et appuie sur OK.

Ton fichier est maintenant ajouté.



Appuie sur OK.

Pour **tester ta séquence**, appuie sur



(Tu dois tester la séquence **avec le vrai robot** pour entendre la voix)

POUR FAIRE BOUGER TON ROBOT NAO



EN MODE ANIMATION

SOURCE : ALDEBARAN DOCUMENTATION, CHOREGRAPHE TUTORIALS

Dans le **mode animation**, c'est toi qui va placer le robot dans la position désirée. Nous souhaitons, dans cet exemple, que NAO se place dans la position du *Penseur*, de Rodin :

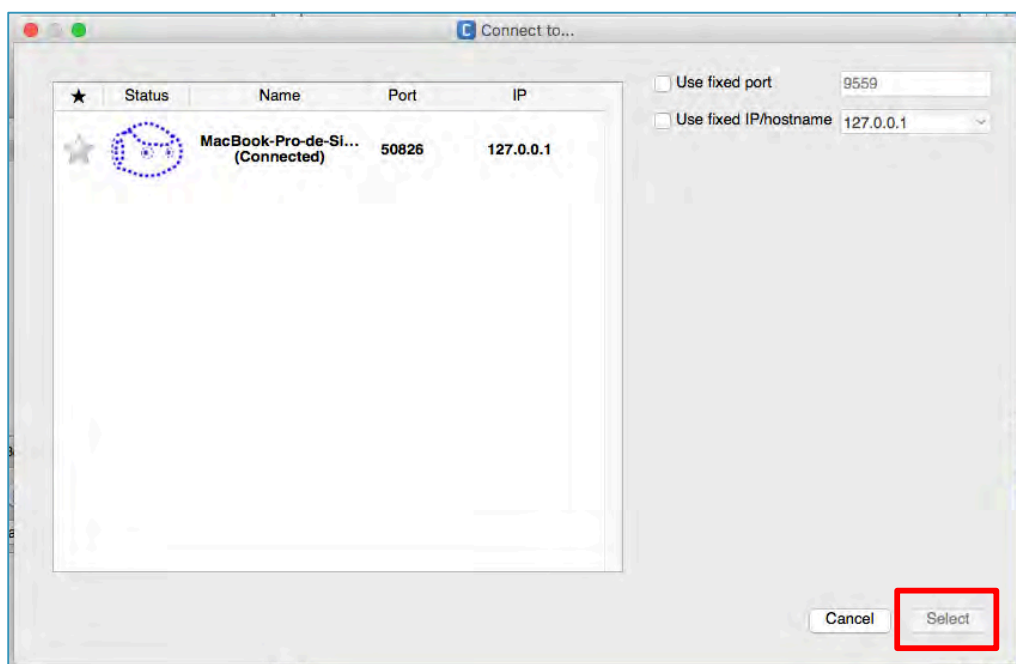


D'abord, assure-toi que tu es connecté à **un vrai robot** (pas le robot virtuel).

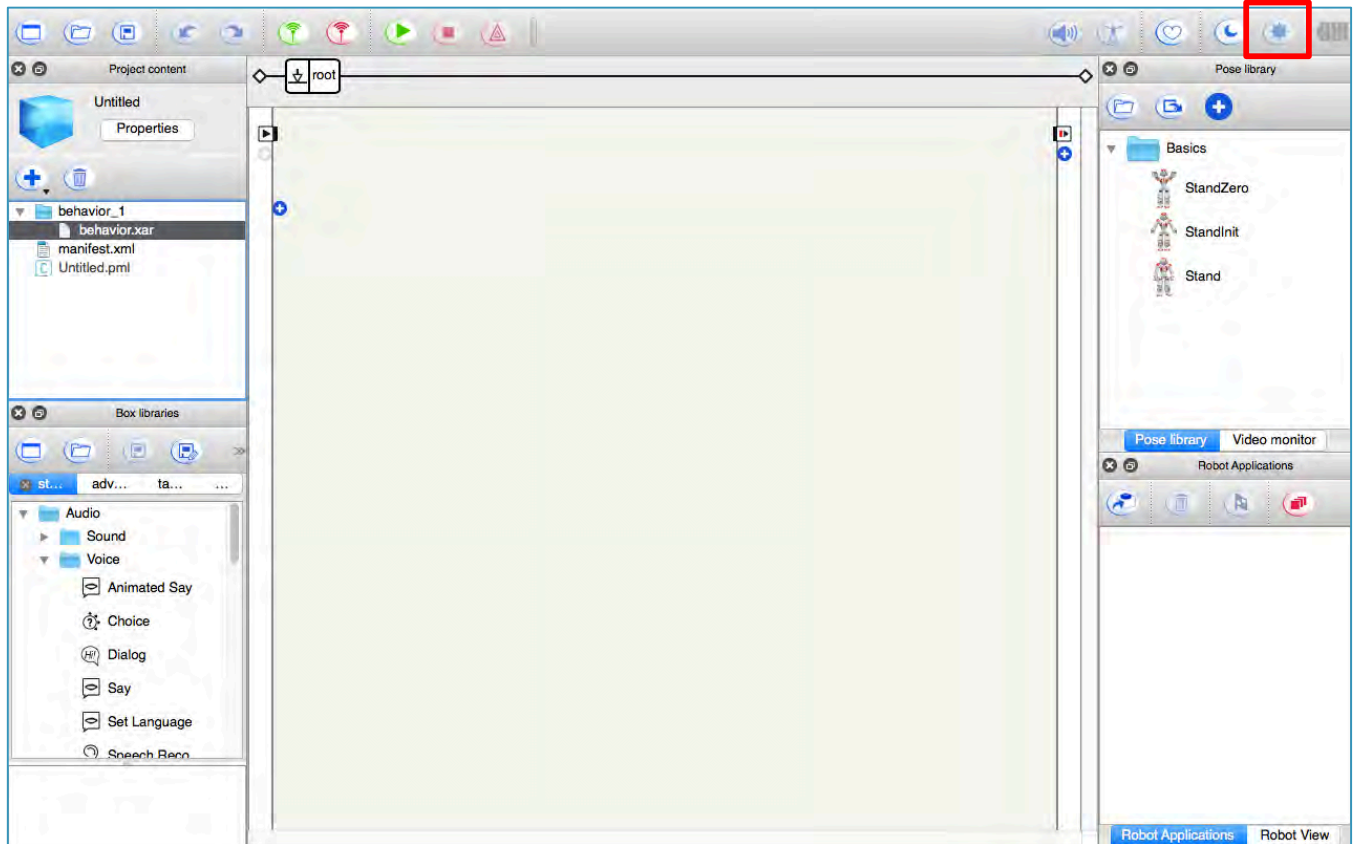
Appuie sur



, sélectionne le bon robot et appuie sur **Select**.

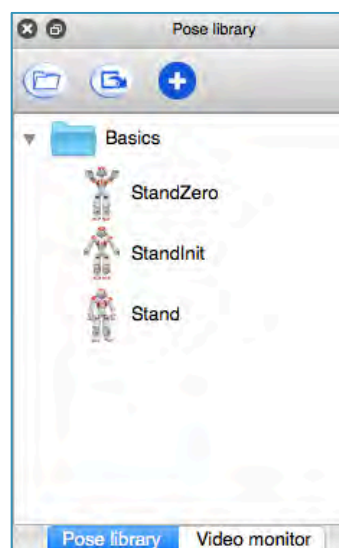


Réveille ton robot en appuyant sur



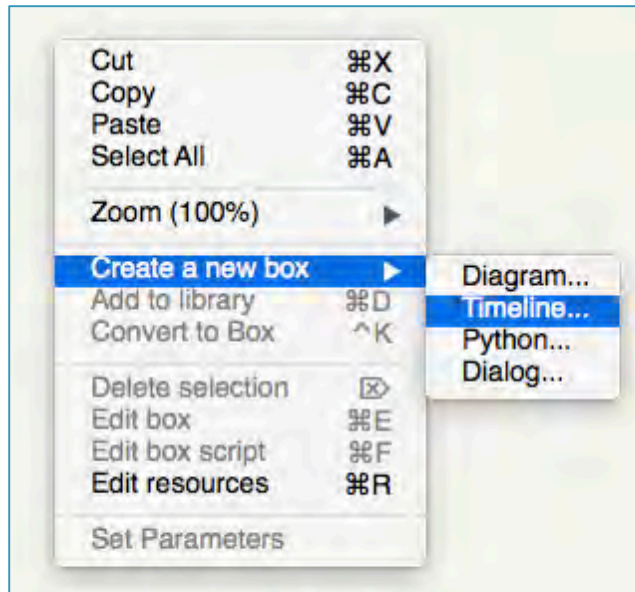
Si tu ne peux pas appuyer sur le bouton, c'est que ton robot est déjà réveillé.

Ton robot devrait être dans la **position StandInit**. S'il ne l'est pas, rends-toi dans la fenêtre *Pose library* et sélectionne la position :

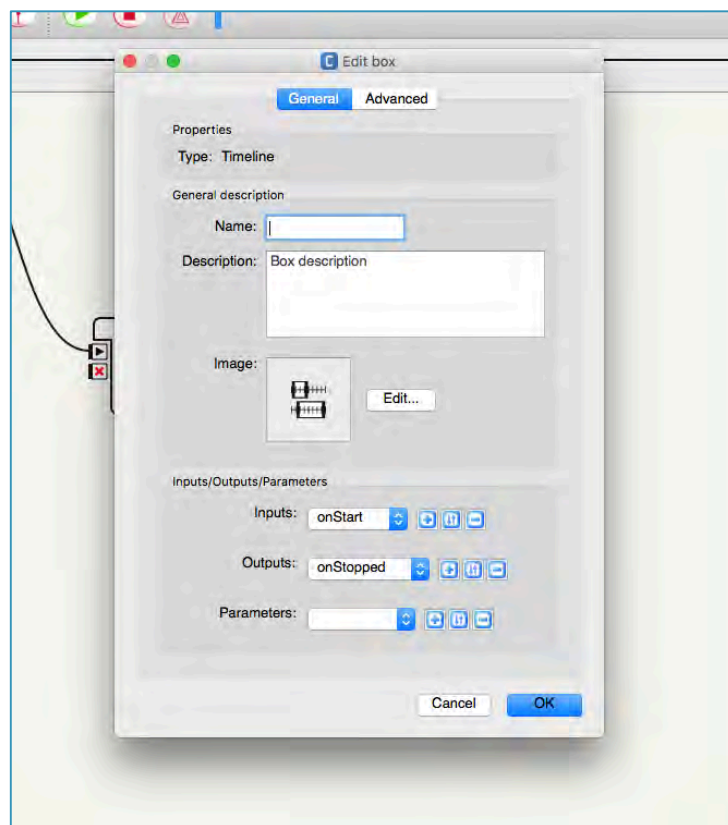


Maintenant, tu dois créer une nouvelle boîte *Timeline*.

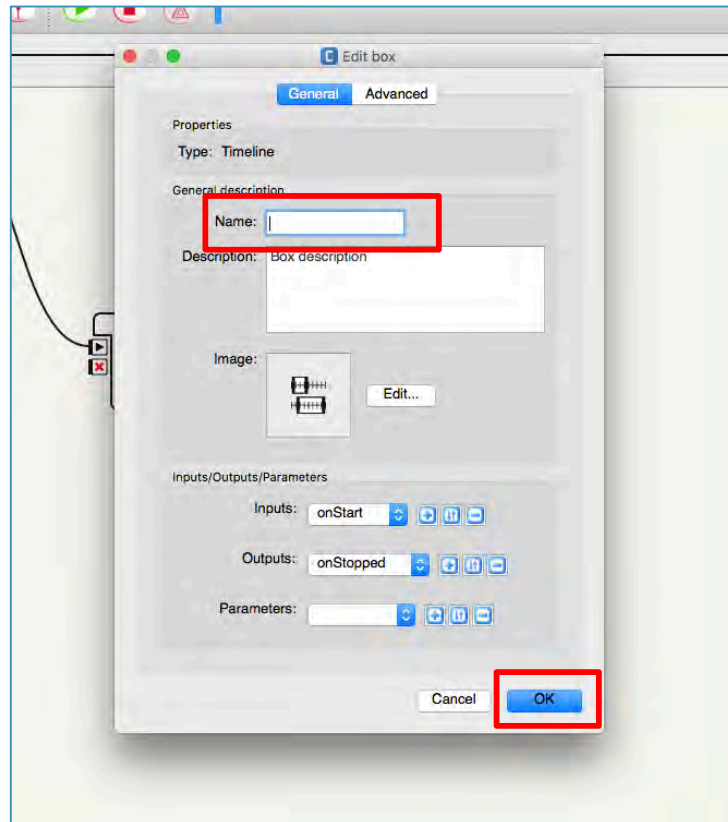
Clique-droit dans ton espace de travail. Sélectionne **Create a new box**, puis, **Timeline...**



La fenêtre *Edit box* apparaît :

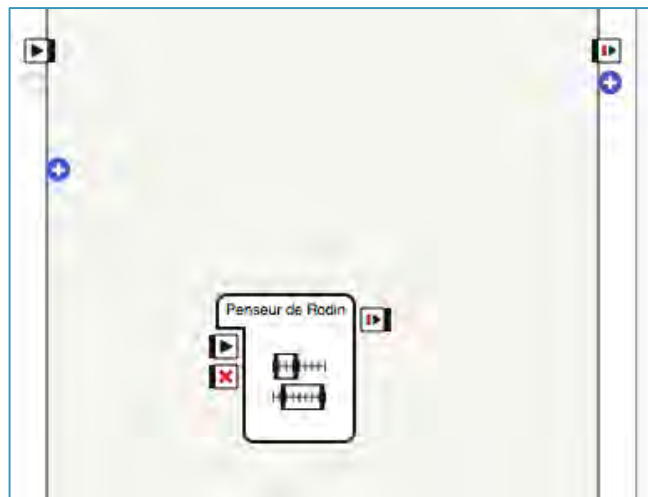


Nomme la boîte *Penseur de Rodin*.



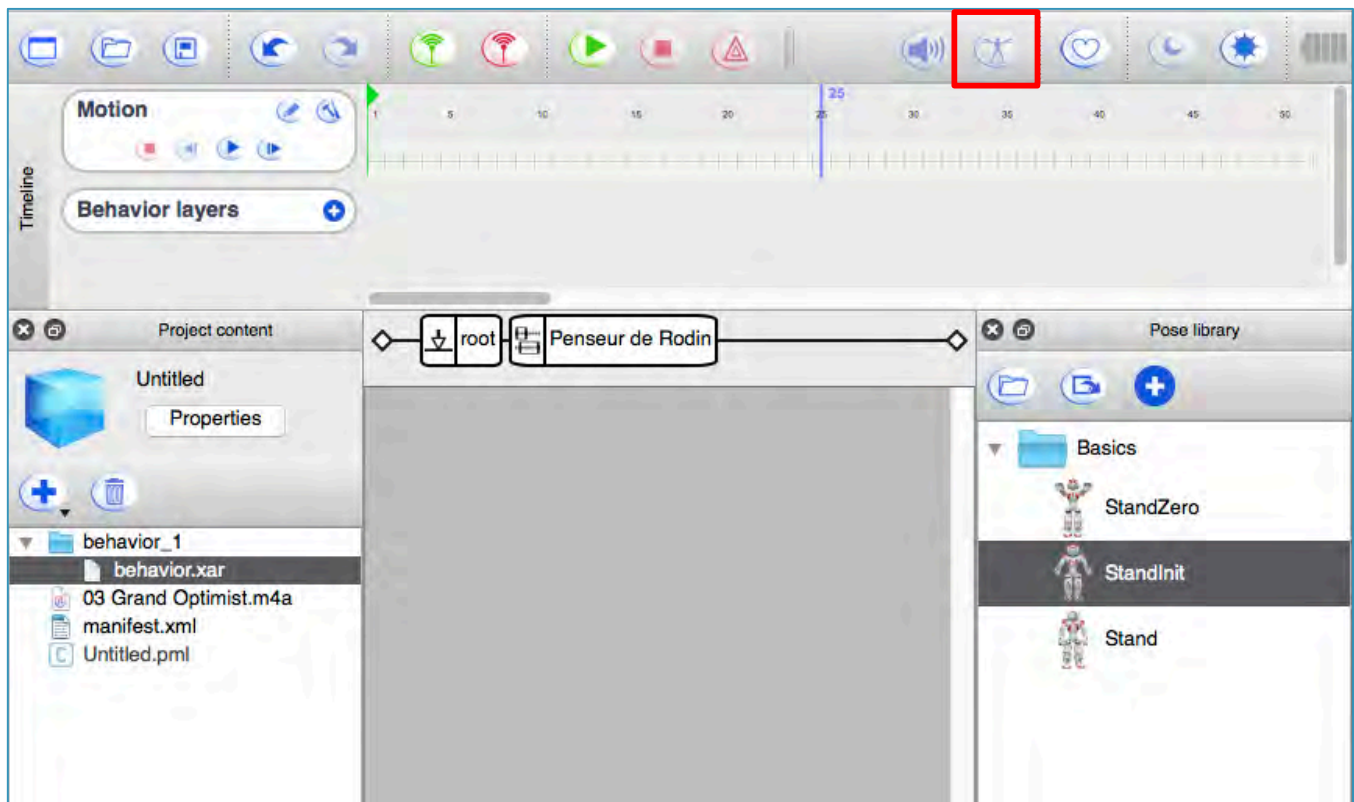
Si tu veux, tu peux ajouter **une description** (ce n'est pas obligatoire).


Quand tu as écrit le nom de la boîte, **appuie sur OK**.




Tu dois maintenant **double-cliquer** sur la boîte.

Tu arrives ici :



Appuie sur le bouton d'animation : 

Le bouton va devenir rouge  , ce qui veut dire que le mode animation est activé.

Pour enregistrer la première position (position de départ), appuie sur le récepteur tactile sur sa tête, avec ta main.

Pour la deuxième position, nous allons décomposer le mouvement :

1. D'abord, **tiens bien** ton robot par la taille/le torse pour éviter qu'il ne tombe.
2. **Appuie sur les deux capteurs** des pieds pour relâcher la tension des jambes.
3. Descend le torse pour faire **plier les jambes** :



4. **Avance le pied droit** et tourne-le légèrement **vers la droite** :



5. **Appuie sur les deux capteurs** des pieds pour rétablir la tension dans les jambes.
6. Pour enregistrer la deuxième position, appuie sur les capteurs sur sa tête.

La prochaine partie du mouvement concerne les bras et la tête de NAO :

7. Appuie sur le capteur tactile au milieu de la tête pour **relâcher la tension de la tête**.
8. **Baisse la tête de NAO vers l'avant** jusqu'à tu ne puisses plus la baisser.
9. Appuie sur le capteur tactile au milieu de la tête pour **rétablir la tension**.
10. **Penche le torse** de NAO vers l'avant.
11. **Assure-toi que le robot est stable !**

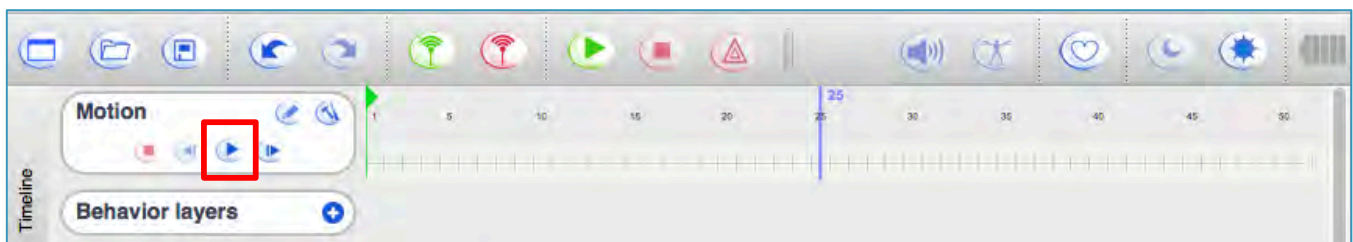


12. Si la position est stable, **enregistre la dernière position** en appuyant sur les capteurs tactiles sur sa tête.

13. Tu peux maintenant **appuyer sur**



Pour **tester le comportement**, appuie sur :



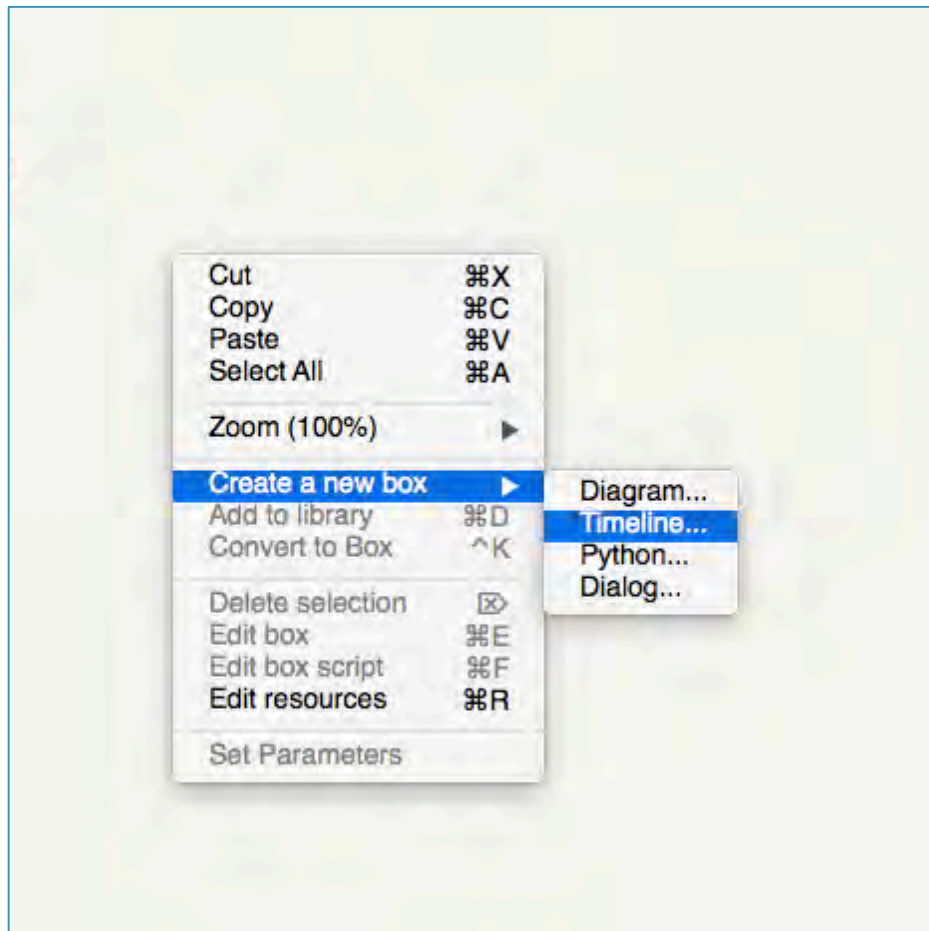
CRÉER UN MOUVEMENT

SOURCE : LE ROBOT NAO - [HTTP://STETIEL2.FREE.FR/AN2016/AFF_RUB.PHP](http://steteil2.free.fr/AN2016/AFF_RUB.PHP)

Il existe plusieurs mouvements (boîtes) déjà enregistrés, mais tu peux aussi en créer toi-même.

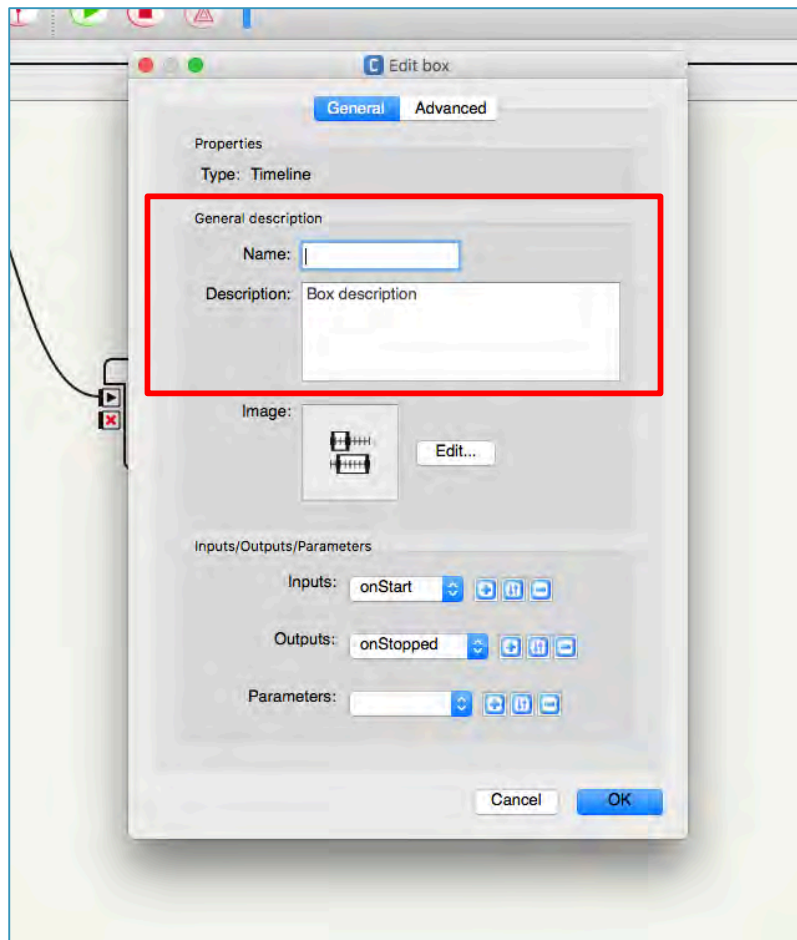
Pour créer une nouvelle boîte, **clique-droit** sur l'espace de travail.

Sélectionne l'option *Create a new box*.

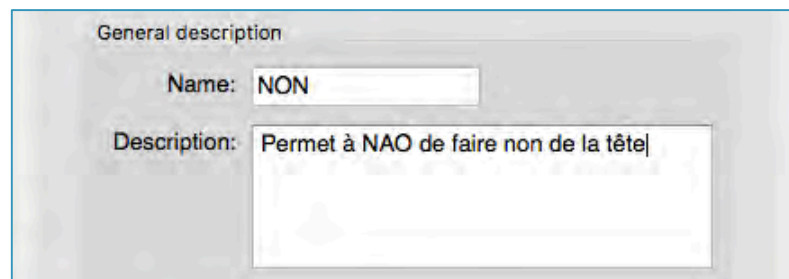


Il y a plusieurs options pour une nouvelle boîte. Choisis l'option *Timeline...*

Une nouvelle fenêtre va apparaître.



Tu dois maintenant **nommer ton comportement** (tu peux aussi ajouter une description). Pour cet exemple, nous allons faire dire non (avec la tête) à NAO :



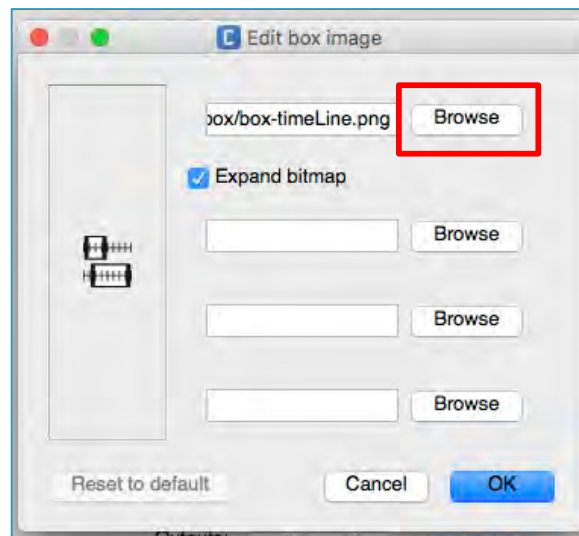
Si tu veux, tu peux aussi **ajouter une image** (ce sera plus facile de reconnaître tes boîtes dans ton espace de travail).

Appuie sur :

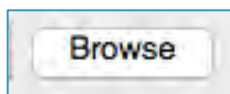


Tu peux créer une image toi-même (sur *Paint*, par exemple), ou la prendre sur Internet.

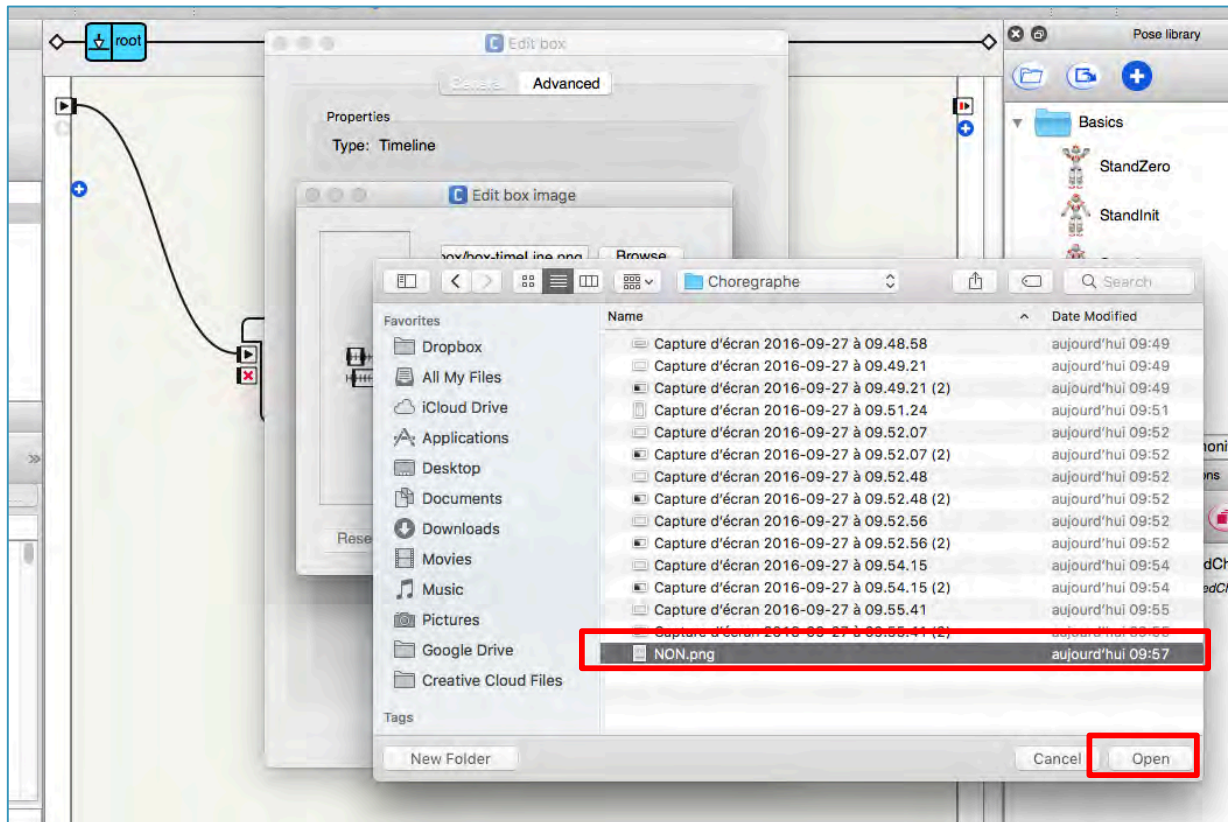
La fenêtre *Edit box image* va apparaître :



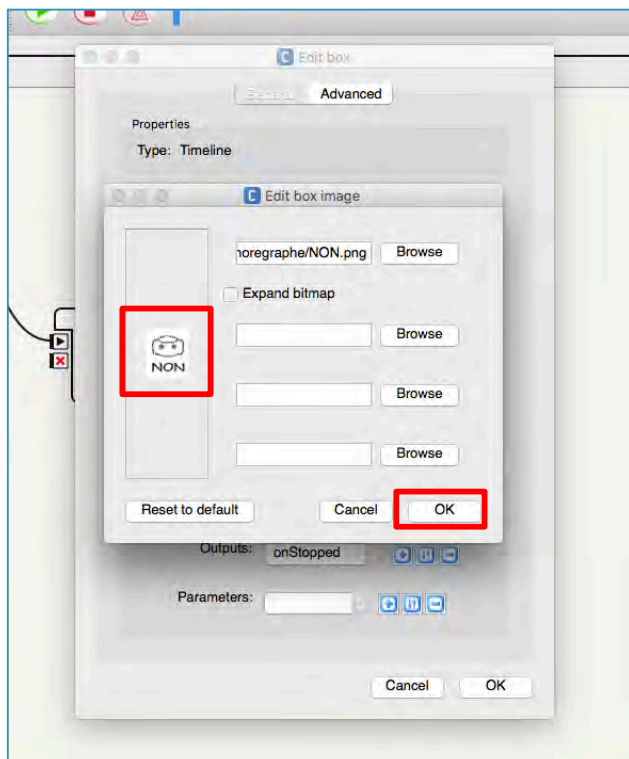
Appuie sur



Cherche dans tes documents l'image que tu as choisie :



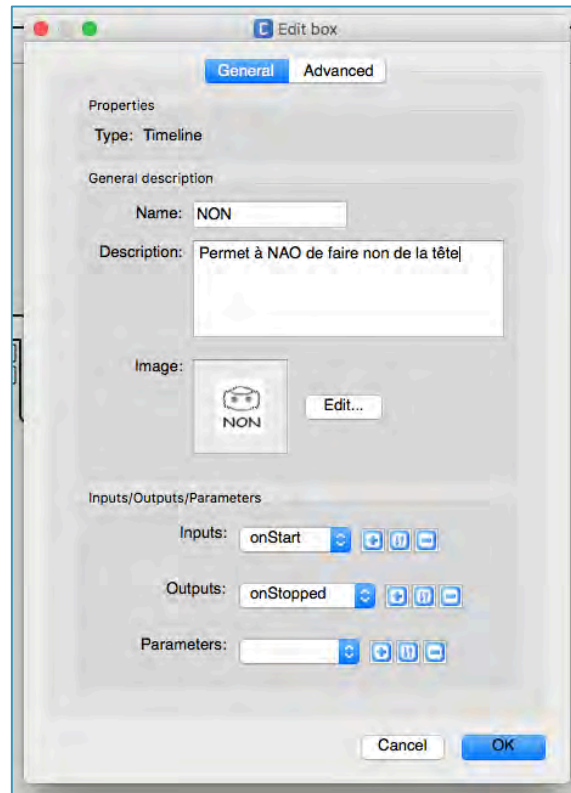
Appuie sur *Open*.



Ton image apparaît à gauche, dans la fenêtre.

Appuie sur OK.

Tu reviens à la fenêtre *Edit box*.



Tu peux maintenant **appuyer sur OK**.

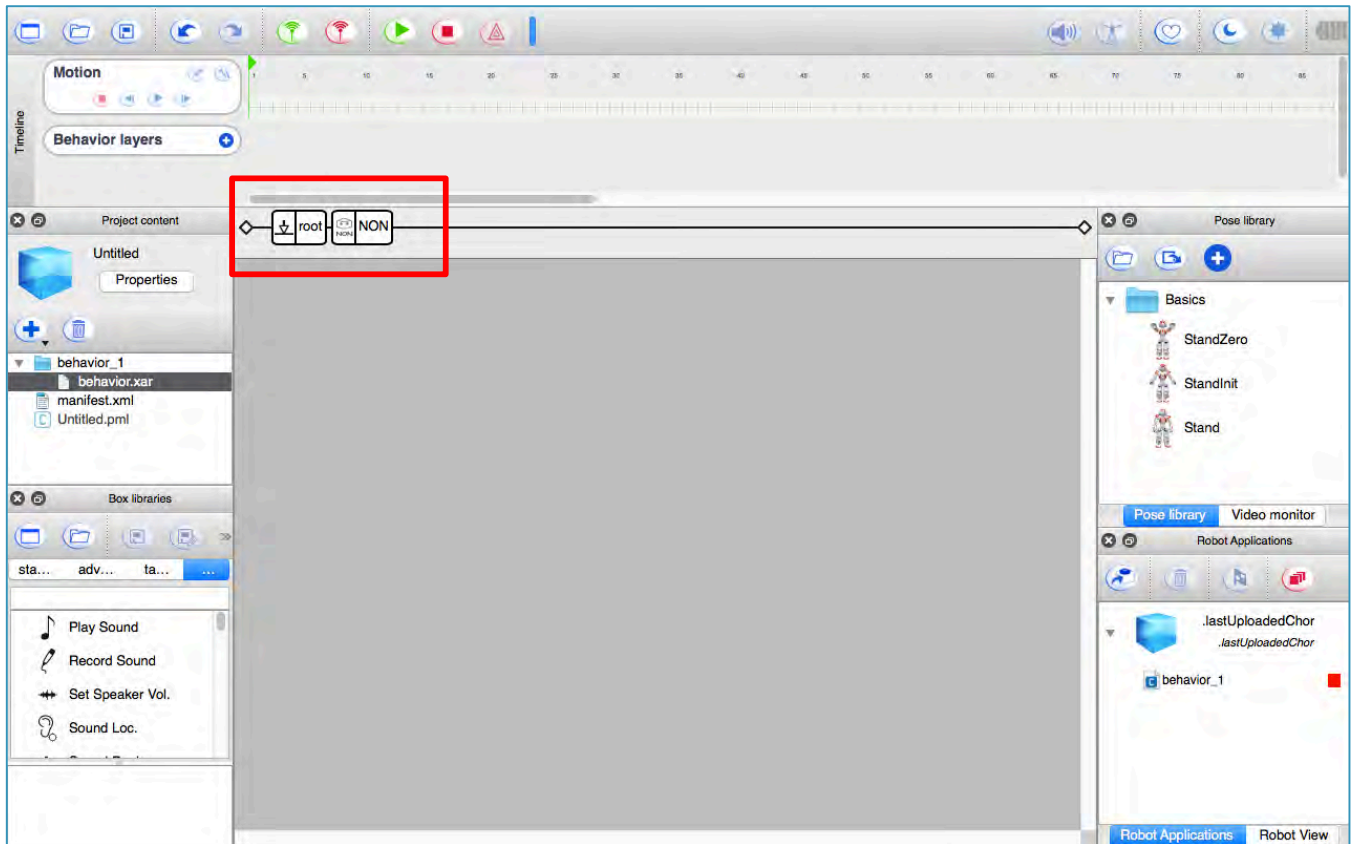


Ta boîte aura donc l'image et le titre que tu lui as donnés.

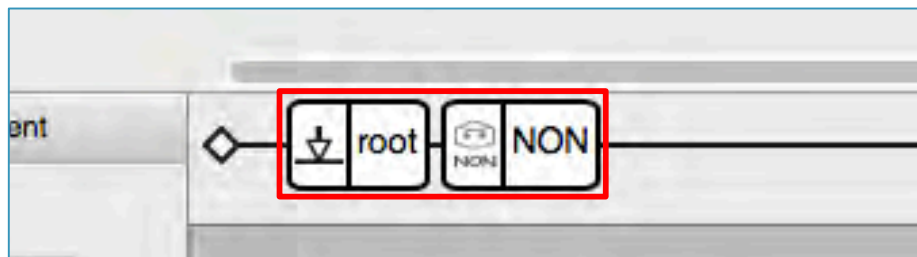
Tu viens donc de créer une boîte. C'est comme l'étiquette du mouvement : maintenant, tu dois dire au logiciel quel est ce mouvement (quelles parties du robot vont bouger et comment)...

Double-clique sur la boîte *NON*.

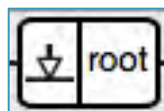
Tu arrives sur cette page :



Tu peux voir que tu es dans la boîte *NON* en regardant ici :

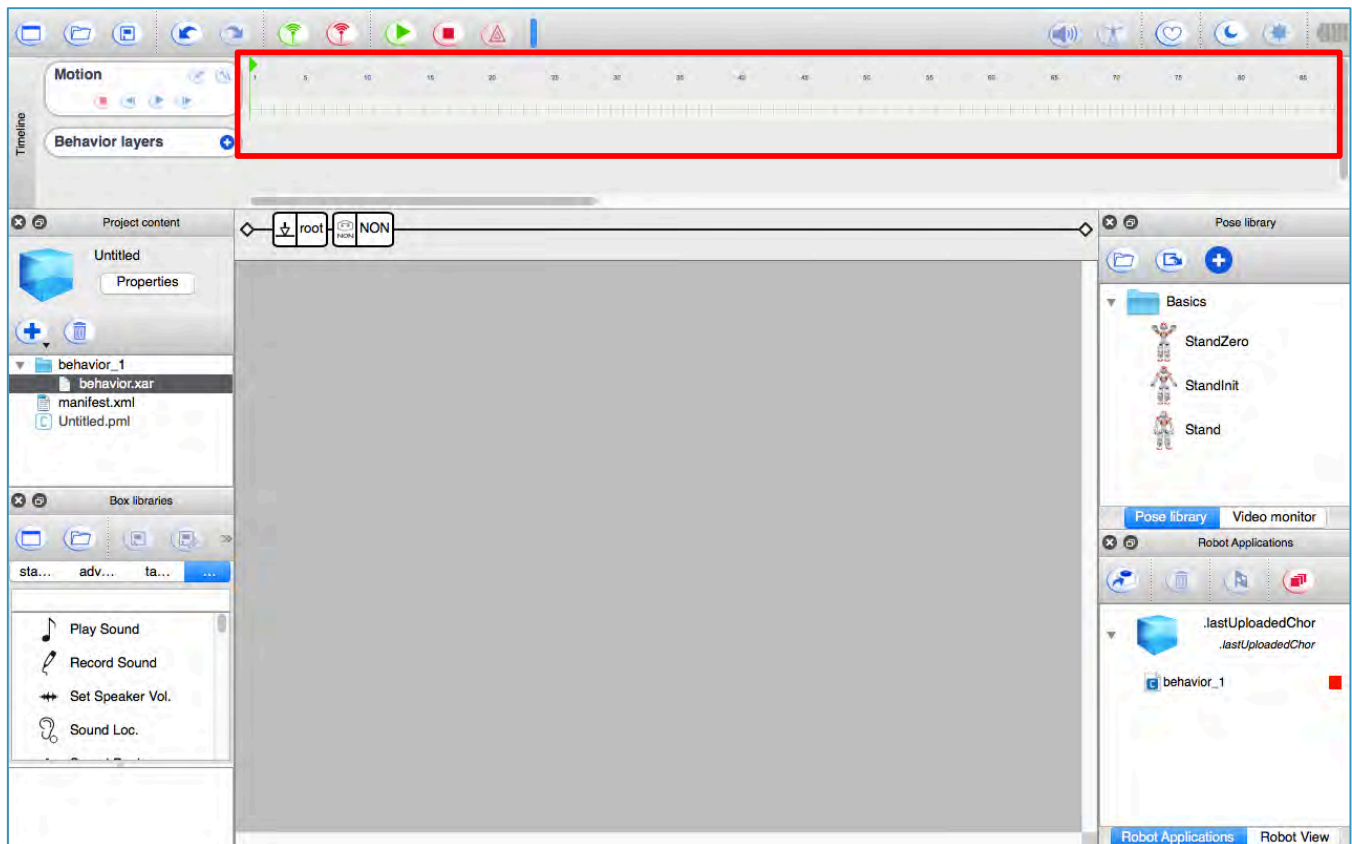


Pour revenir à ton **espace de travail principal**, appuie sur :



Tu vas maintenant créer les différentes positions de ton mouvement *NON*.

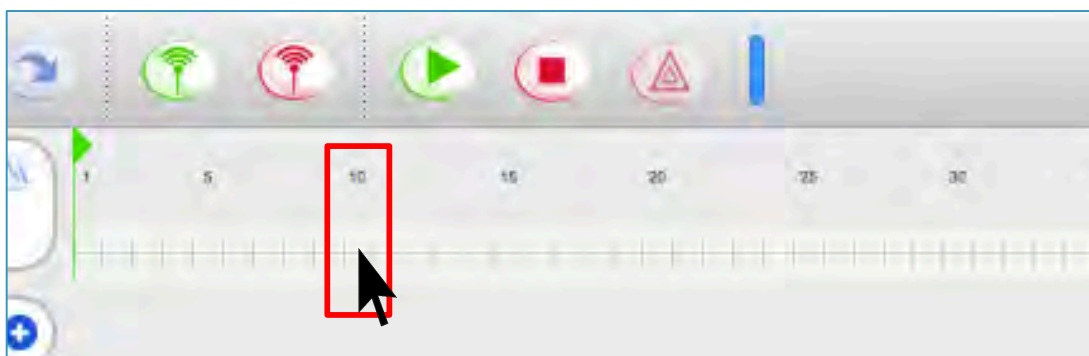
Dans le haut de la fenêtre, tu as ta ligne du temps (la *timeline*).



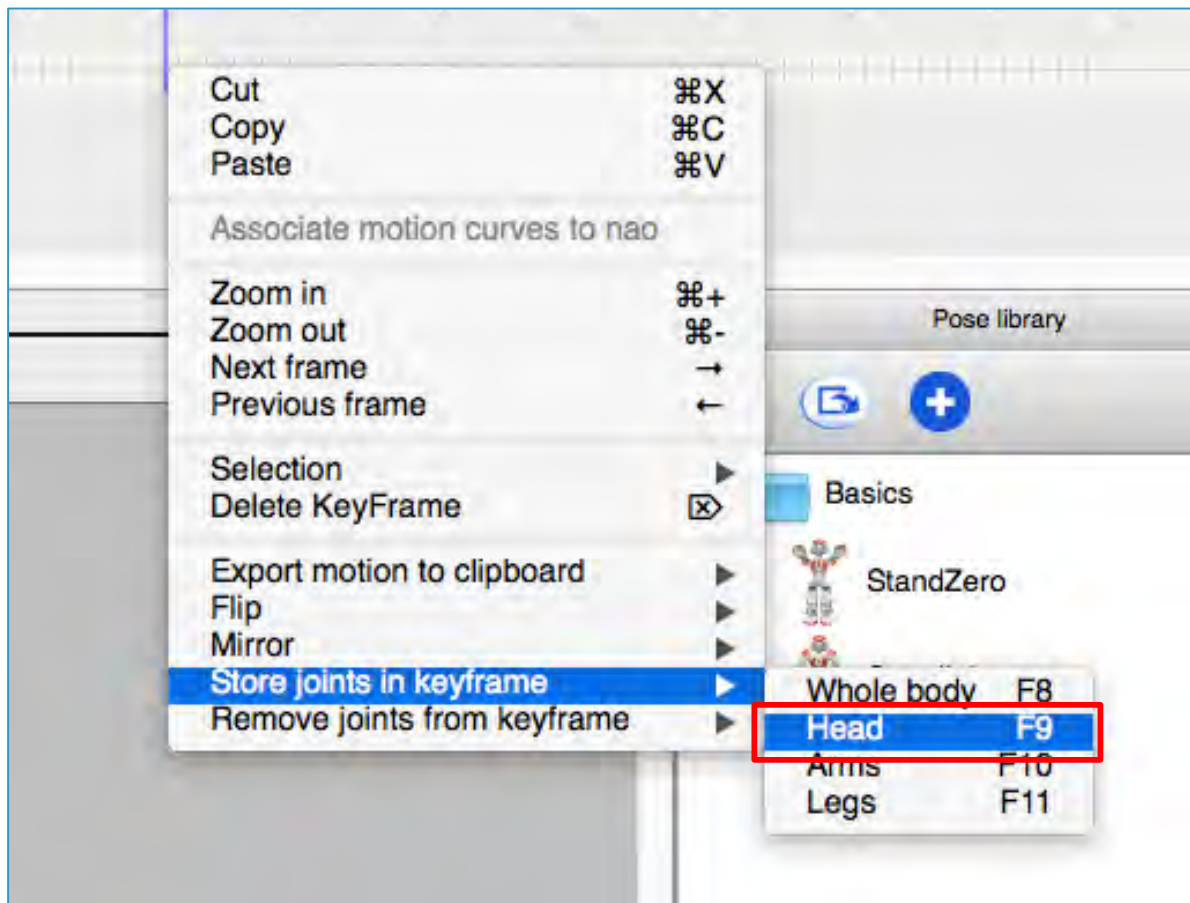
Tu dois créer des marques pour chaque position. Pour créer une marque, **place ton curseur** sur ta ligne du temps.

Nous allons créer une première marque à 10.

Place ton curseur sur le 10 :



Clique-droit et sélectionne *Store joints in keyframe*, puis, **sélectionne** *Head*.

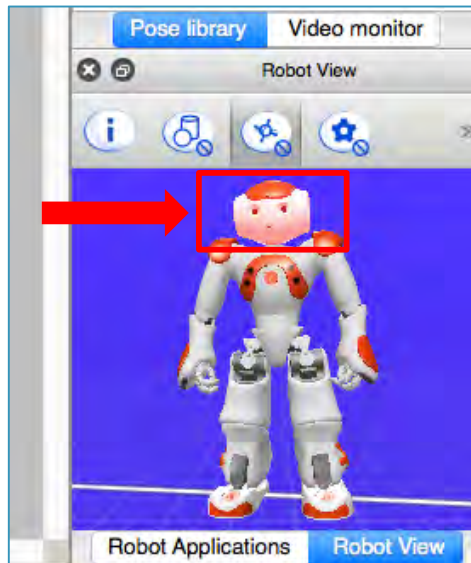


Tu vas voir apparaître un petit rectangle gris foncé sur ta ligne du temps : c'est ta marque.

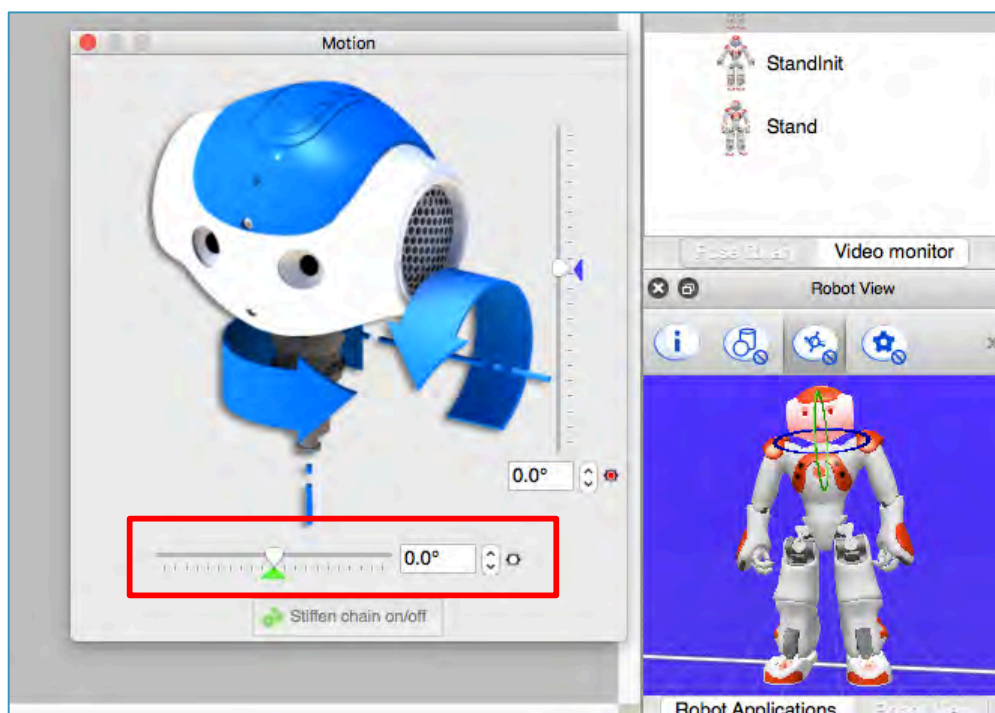


! Assure-toi que ton repère (ligne mauve) est bien situé sur 10, puis,

Clique sur la tête du robot, dans la fenêtre *Robot View* :



La fenêtre *Motion* va apparaître :



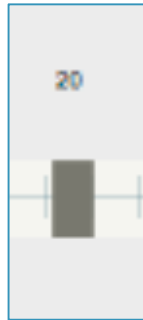
Ici on voit que les curseurs à l'horizontale et à la verticale sont à 0,0 degré.

Cela veut donc dire qu'à 10 (ta marque), la tête du robot sera dans cette position.

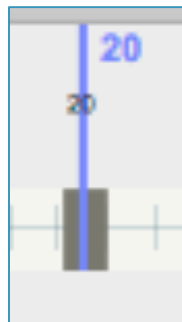
Nous voulons que la tête soit normale à 10, donc on ne change rien.

Ferme la fenêtre *Motion*.

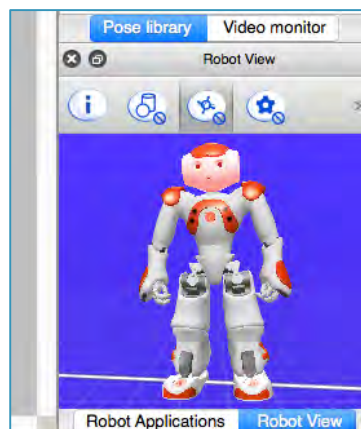
Maintenant, retourne dans ta ligne du temps et **crée une nouvelle marque à 20** en suivant la même procédure.



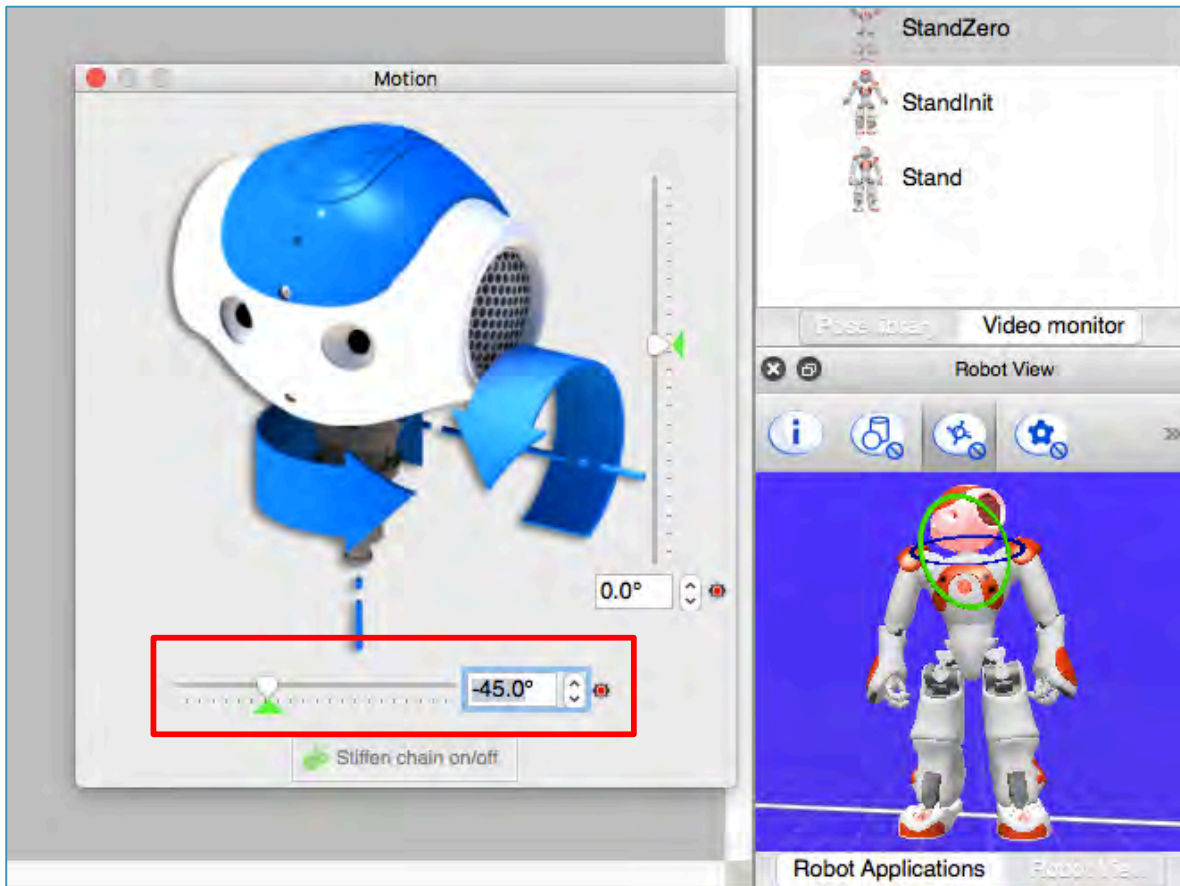
Assure-toi que **ton repère** (ligne mauve) est bien sur ta marque à 20.



Appuie sur la tête dans la fenêtre *Robot View*.



Tu dois maintenant **placer la tête du robot à -45 degrés**, en tirant le curseur vers la gauche ou en utilisant les petites flèches :

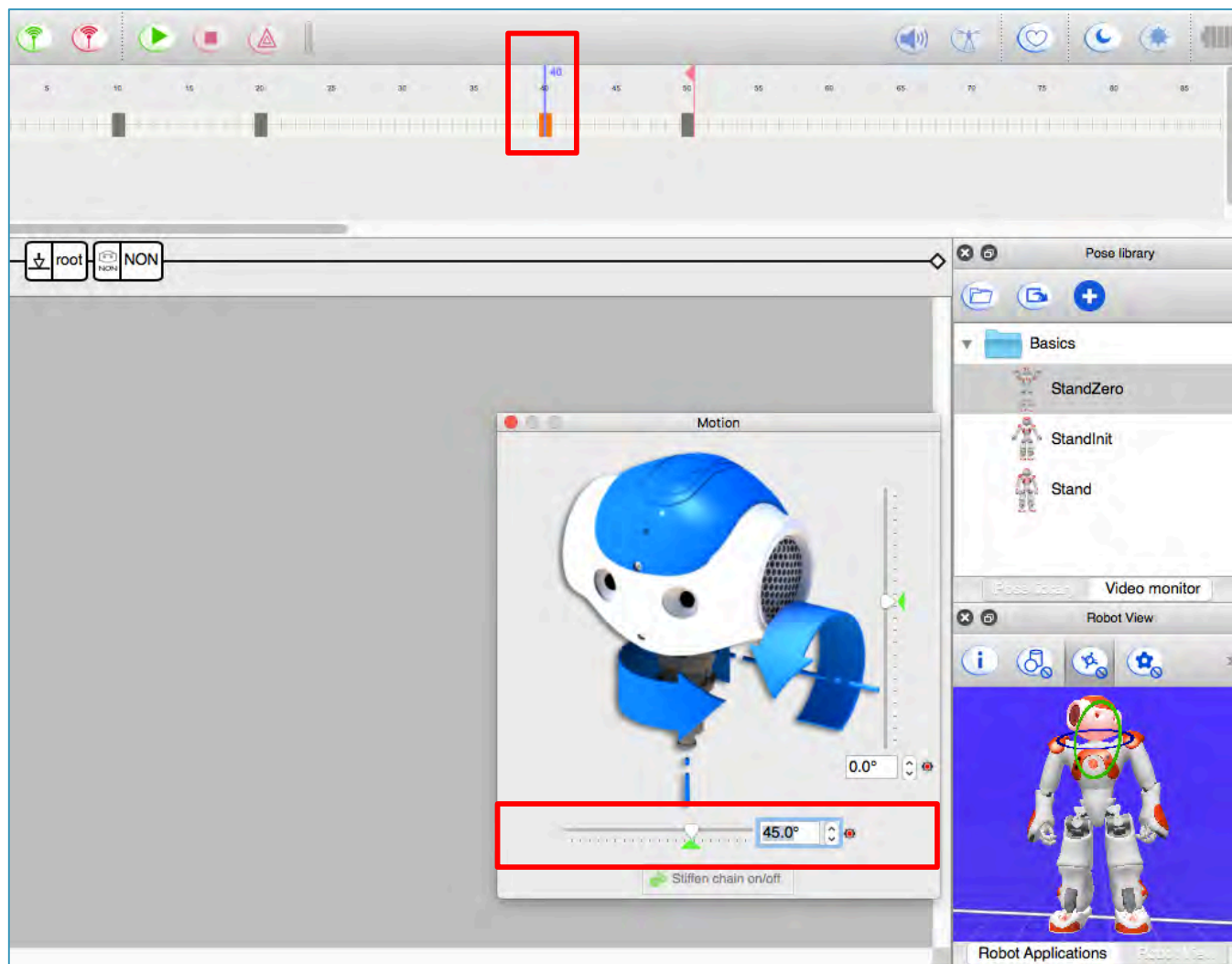


Quand le curseur est bien placé (-45 degrés), **ferme la fenêtre *Motion***.

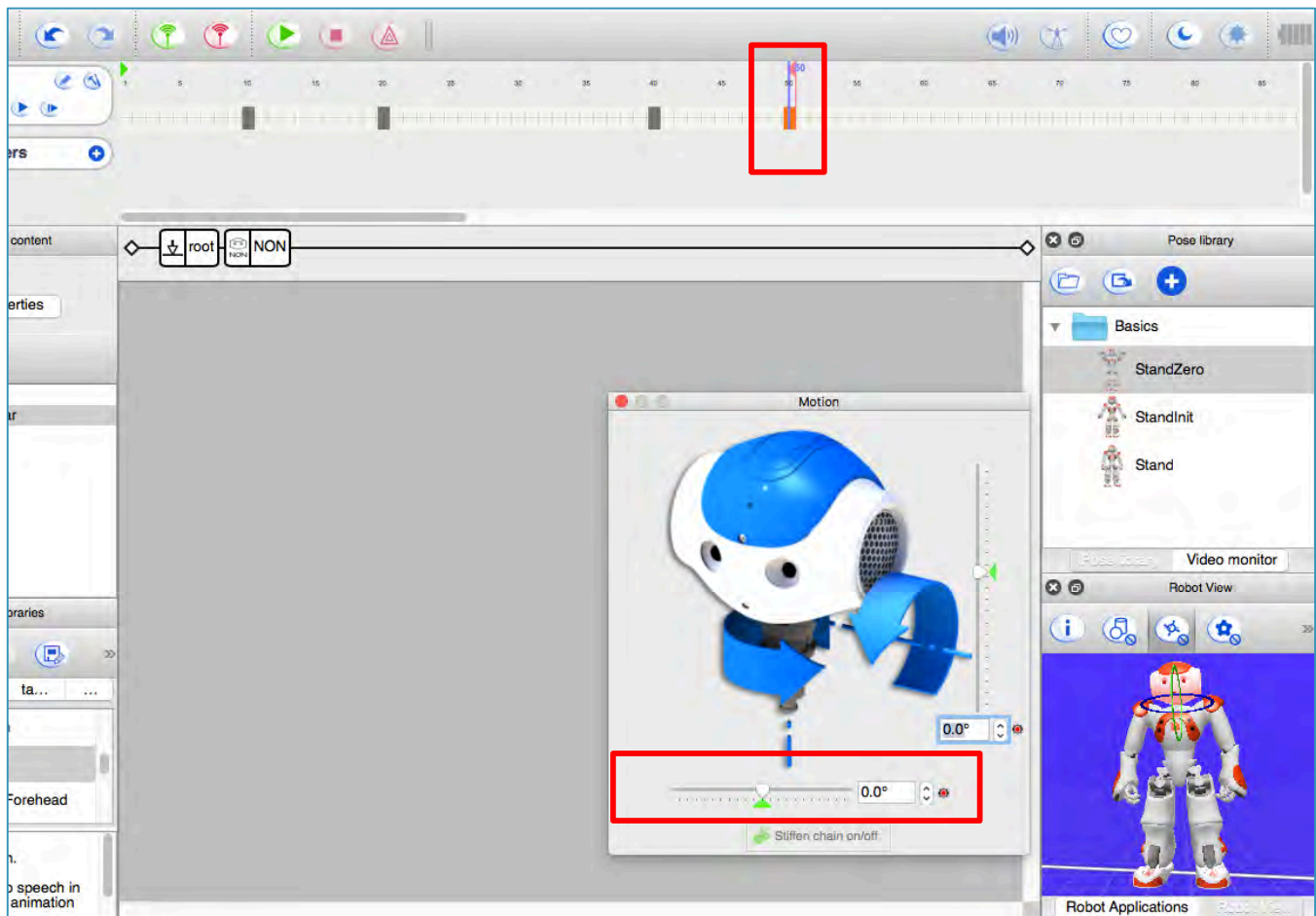
Donc à 10, la tête est à 0 degré et à 20, elle est à 45 degrés.

Tu vas maintenant faire la même opération pour **deux autres marques** :

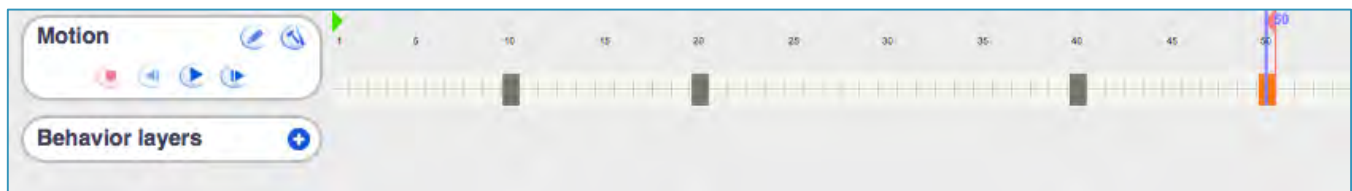
1. **À 40**, la tête du robot doit être à **45 degrés** :



2. À 50, la tête du robot doit être revenue à 0 degré :

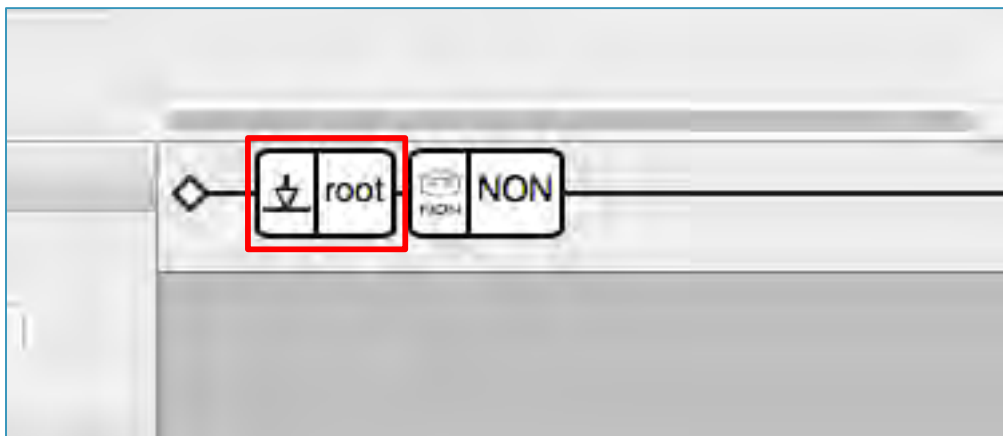
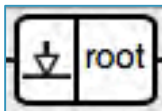


Tu devrais donc avoir 4 marques dans ta ligne du temps :

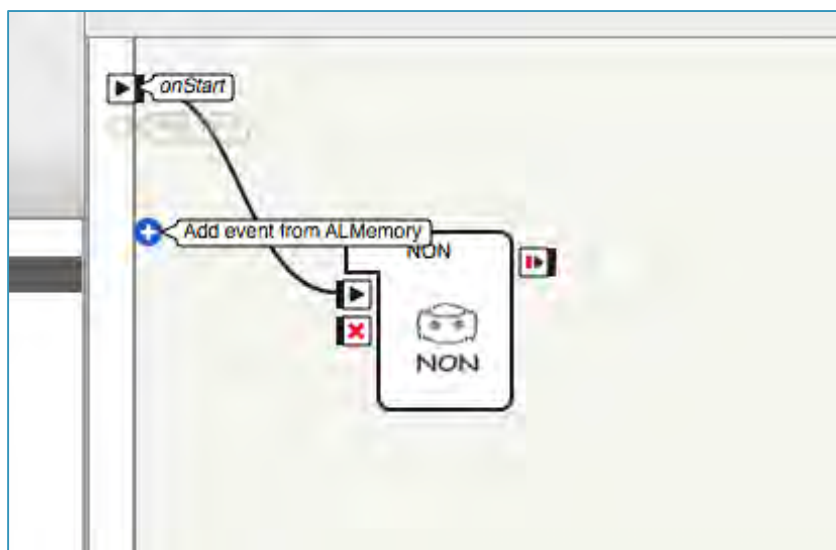


Tu dois maintenant **revenir à ton espace de travail principal**.

Appuie sur



Relie ta boîte *NON* au point de départ de la séquence (*onStart*).



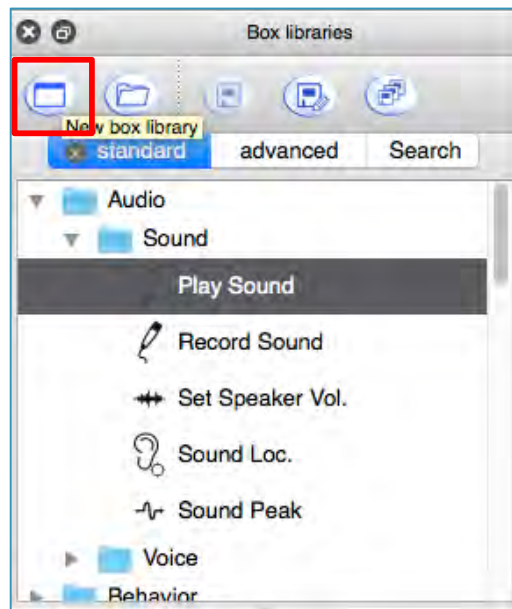
Pour **tester ta séquence**, appuie sur



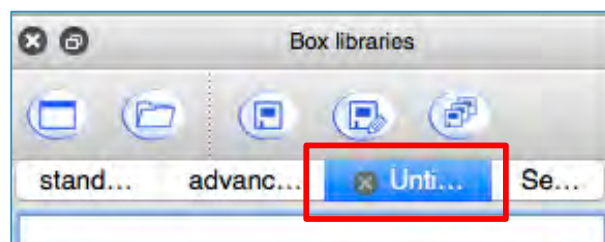
Quand tu crées un nouveau comportement (nouvelle boîte), tu peux l'enregistrer.

Mais tu dois **d'abord créer une nouvelle librairie** :

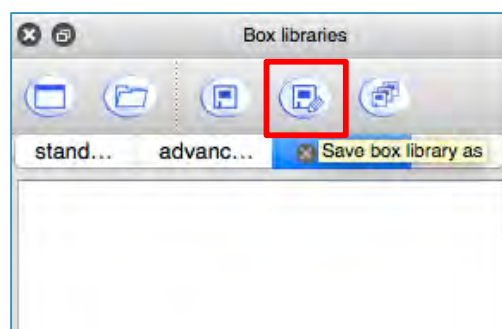
Dans la fenêtre *Box libraries*, appuie sur



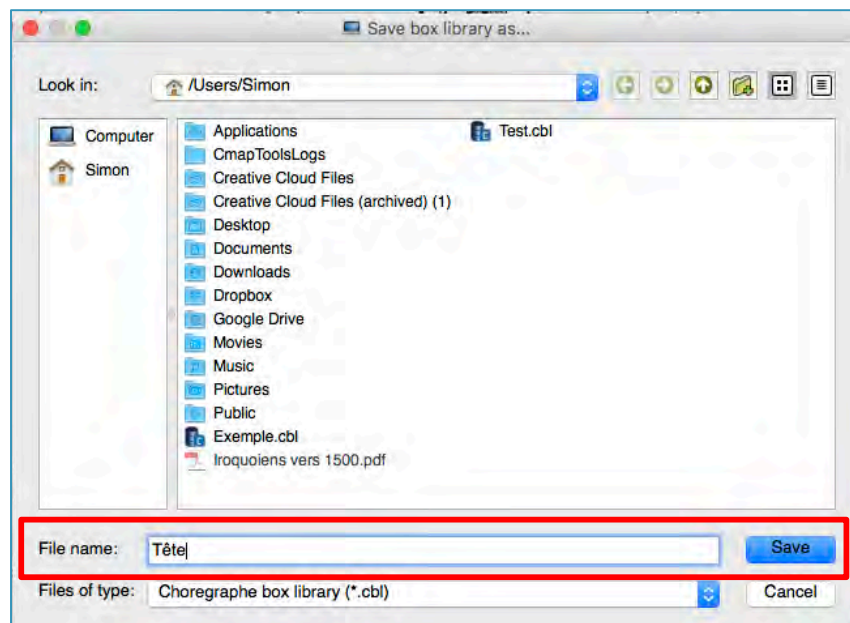
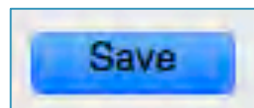
Un nouvel onglet va apparaître :



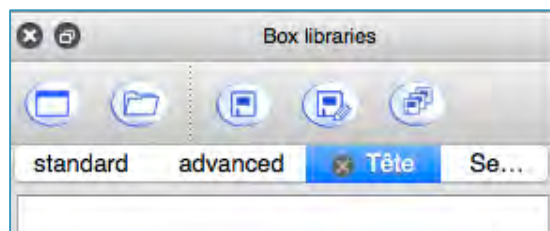
Tu peux **renommer ta nouvelle librairie**. Pour cela, appuie sur :



Écris le nom de ta nouvelle librairie et appuie sur

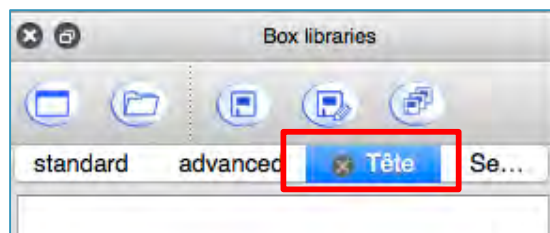


Ta nouvelle librairie est maintenant nommée *Tête* :

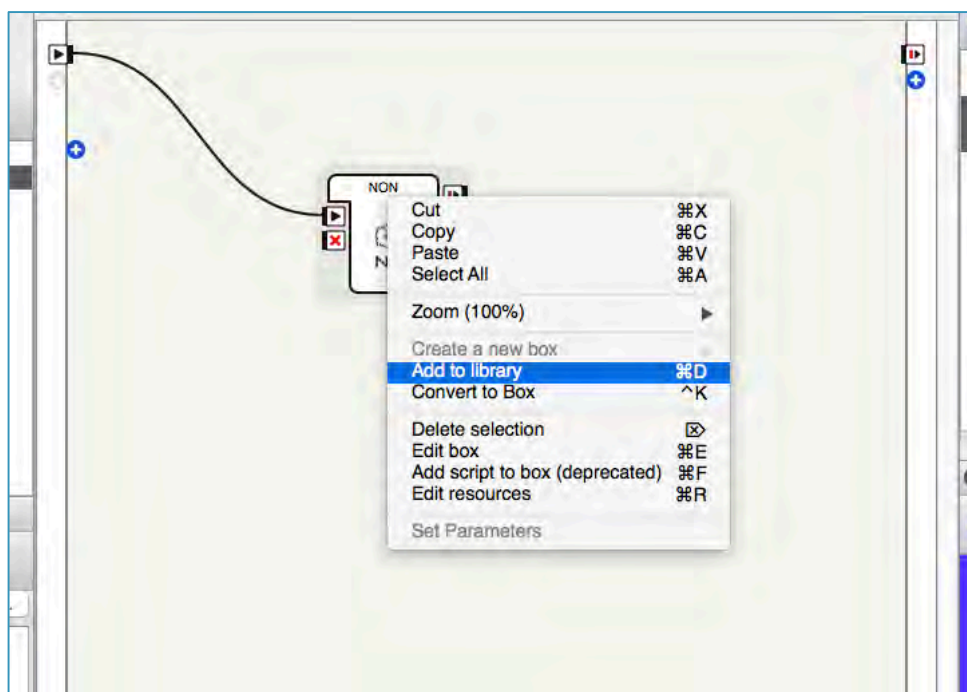


Tu peux maintenant enregistrer ton mouvement (boîte) *NON* dans ta librairie.

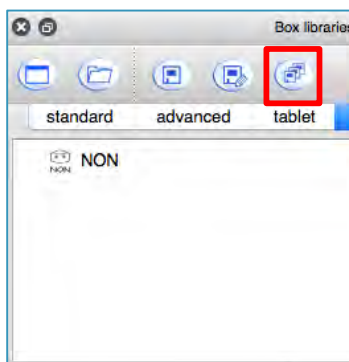
D'abord, assure-toi que ta librairie *Tête* est sélectionnée (en bleu) :



Clique-droit sur la boîte dans ton espace de travail et **sélectionne** *Add to library* :



Ta boîte va maintenant se retrouver dans ta librairie :



Pour enregistrer les changements dans ta librairie,

appuie sur



LE LOGICIEL CHOREGRAPHE

PARTIE 2

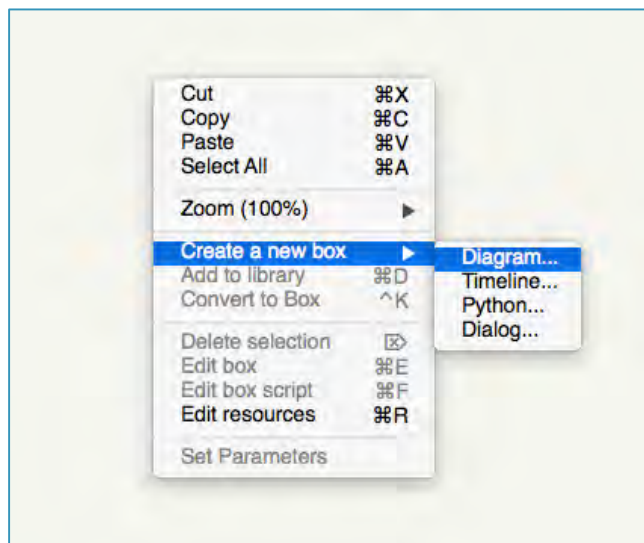
COMMENT CRÉER UN COMPORTEMENT

SOURCE : LE ROBOT NAO - [HTTP://STETAIL2.FREE.FR/AN2016/AFF_RUB.PHP](http://steteil2.free.fr/AN2016/AFF_RUB.PHP)



CRÉER UNE BOÎTE

Clique-droit dans ton espace de travail :



Sélectionne l'option *Diagram*

Quelle est la différence entre *Diagram* et *Timeline*?

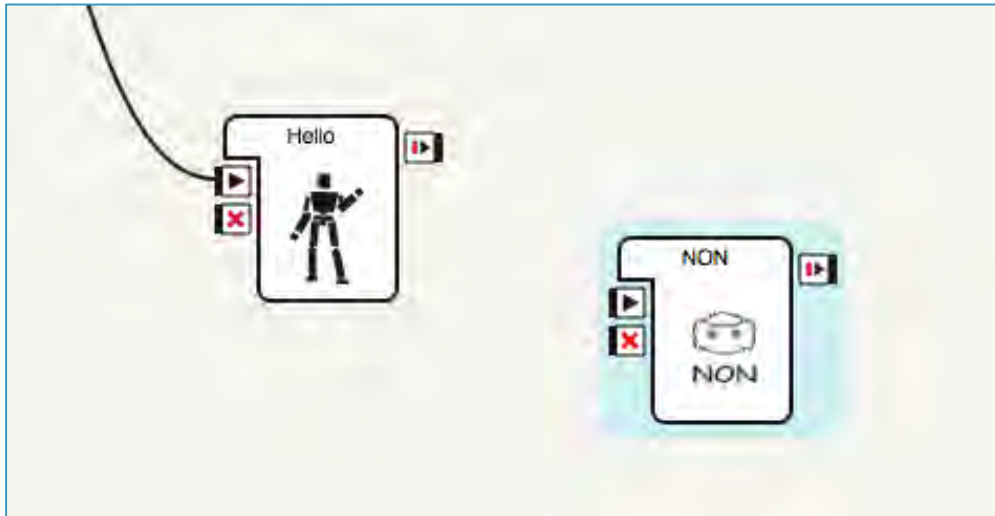
Avec l'option **Diagram**, tu peux créer des comportements en **plaçant les boîtes**, comme tu le fais dans ton espace de travail principal:



Avec l'option **Timeline**, tu dois utiliser **ligne du temps** et insérer des **marques** pour créer un comportement:



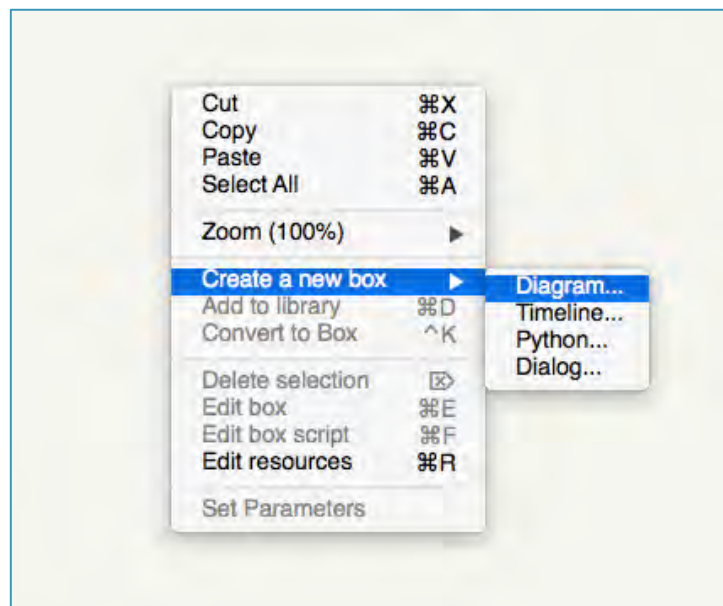
Option *Diagram* :



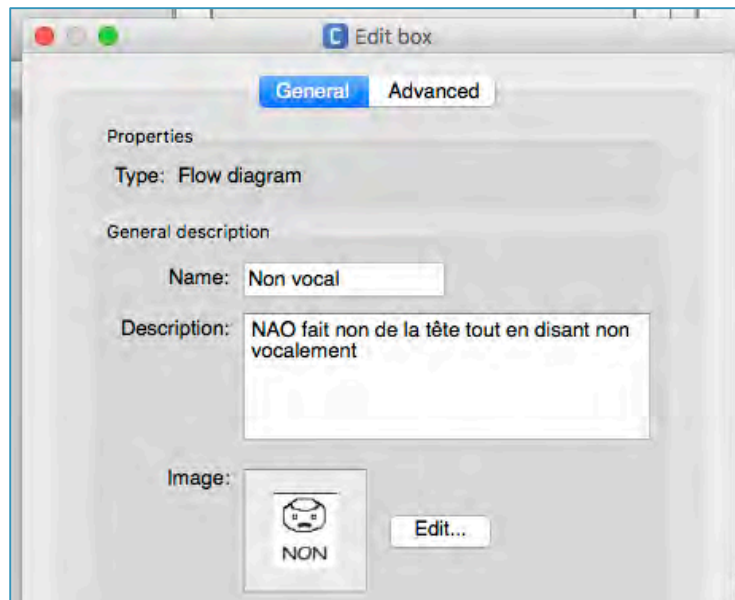
Option *Timeline* :



Pour l'exercice que nous allons faire, nous avons besoin d'une boîte de type *Diagram*.



Quand la fenêtre *Edit box* apparaît, entre les informations suivantes :



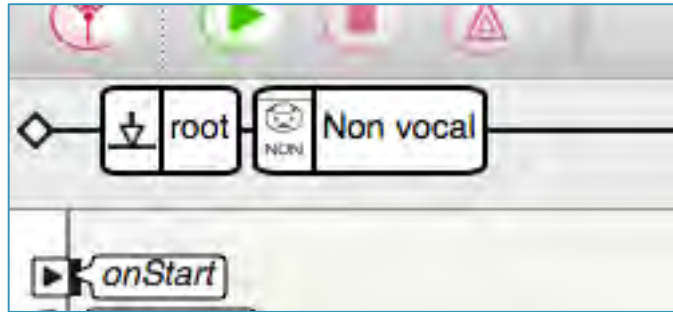
ATTENTION

Tu dois créer l'image toi-même. Tu peux la faire sur un logiciel de dessins comme *Paint*, ou la prendre sur Internet.

Ta nouvelle boîte va apparaître dans ton **espace de travail**.



Double-clique sur la boîte *Non vocal* lumineux :



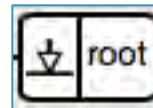
Si une nouvelle case apparaît à côté de la case *root*, ça veut dire que tu es **à l'intérieur** de cette nouvelle case (*Non vocal*).



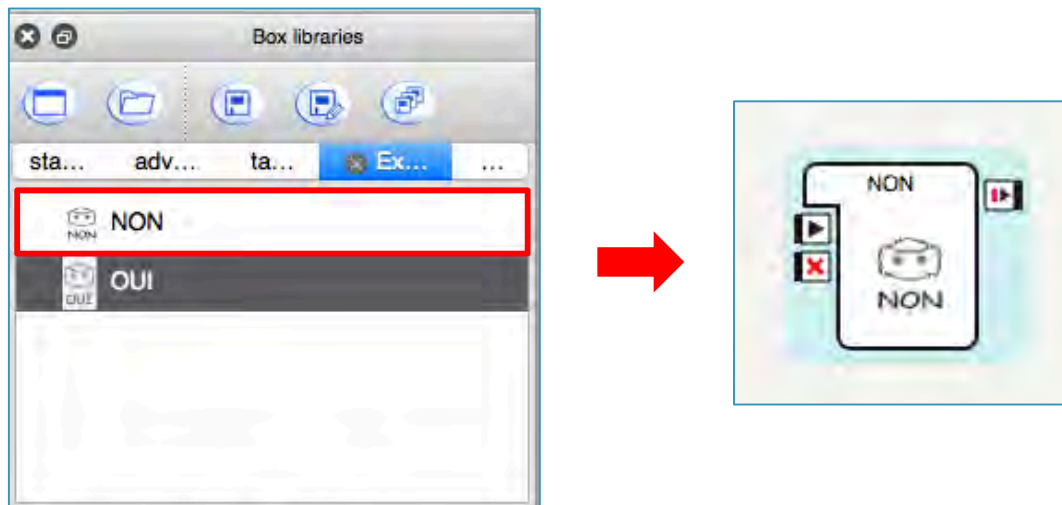
Tu **n'es plus** dans ton espace de travail principal : tu es dans un **espace de travail secondaire**.



Pour retourner dans ton **espace de travail principal**, appuie sur

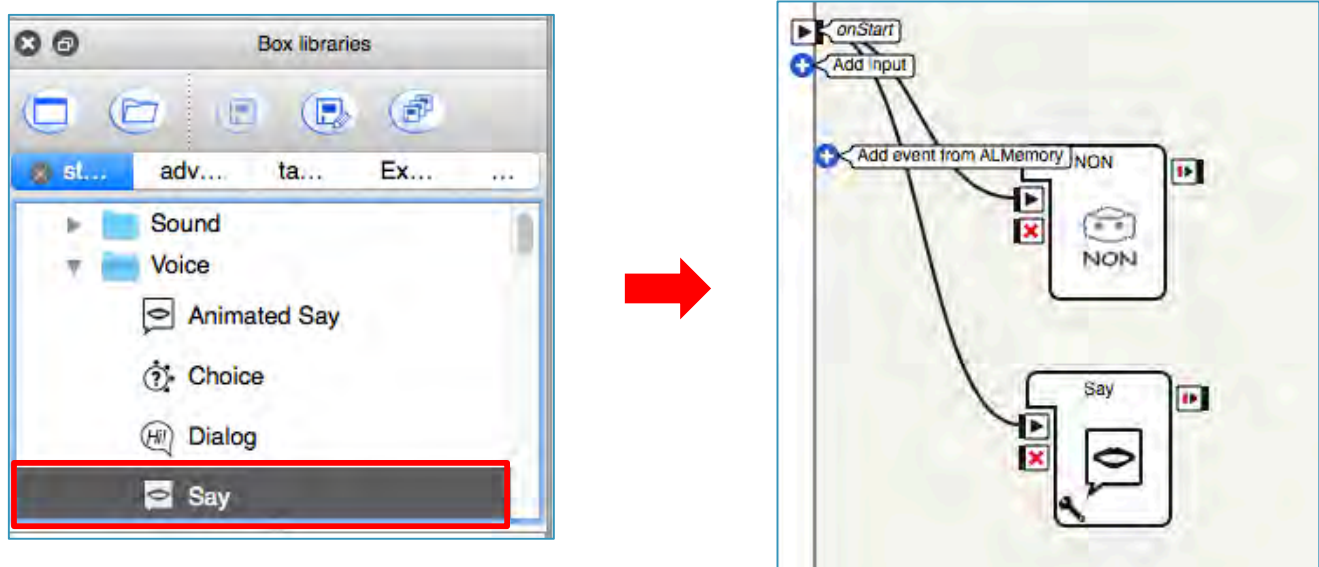


Dans la fenêtre *Box library*, cherche l'onglet que tu as créé dans la *Partie 1* et cherche la boîte *NON* :



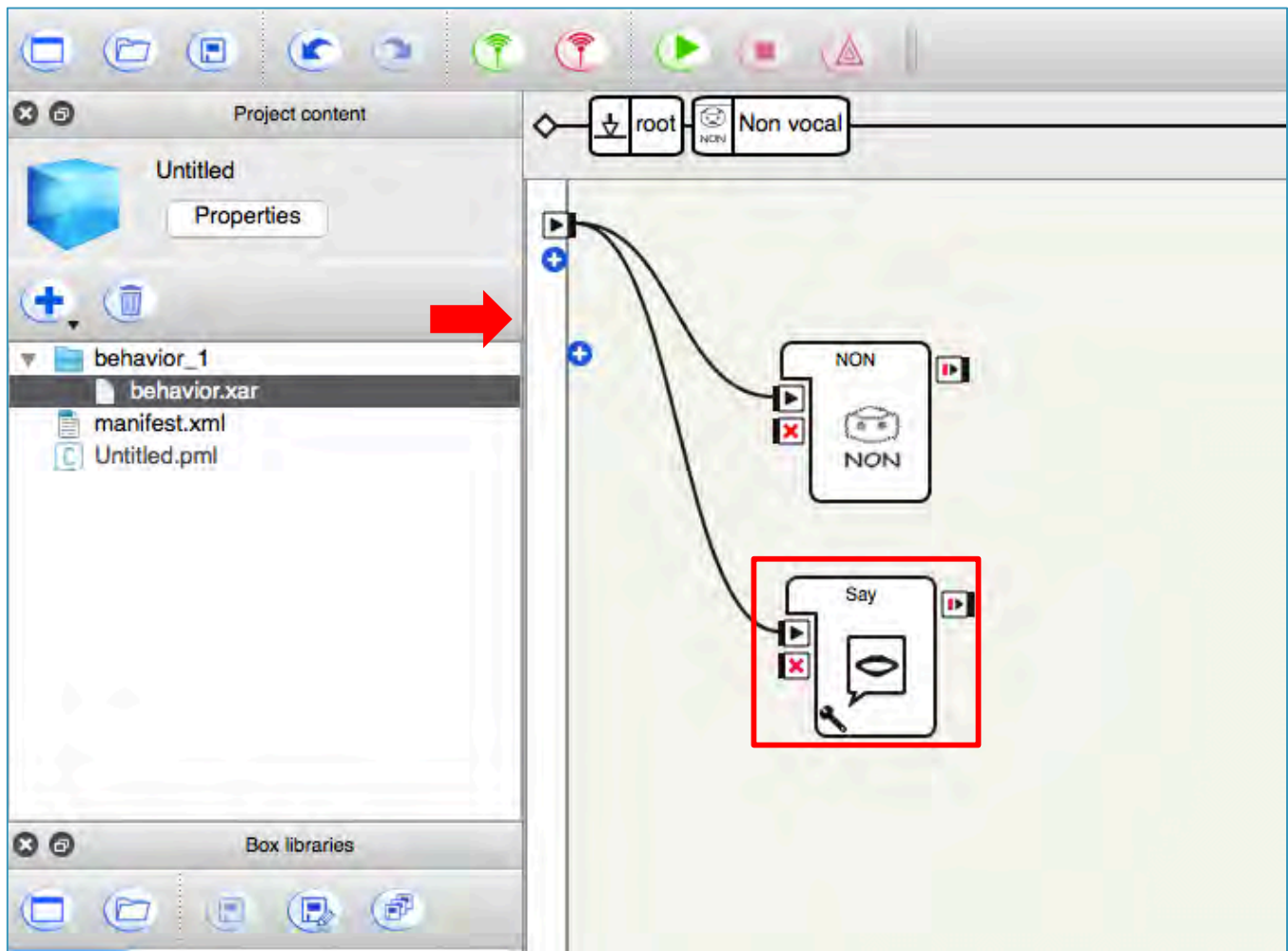
Transporte la boîte *NON* dans ton espace de travail secondaire.

Encore dans la fenêtre *Box library*, dans le premier onglet, cherche le dossier *Voice*, puis *Say* :

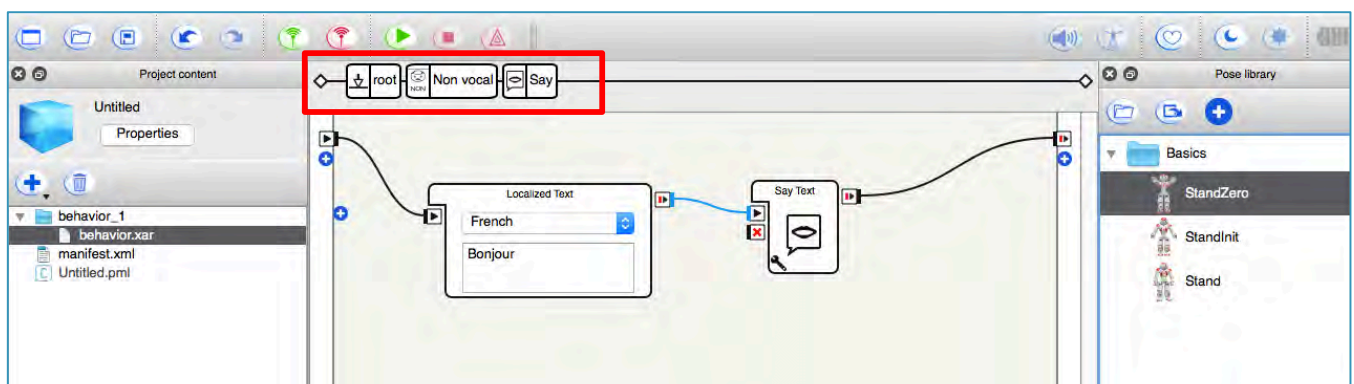


Tu peux **relier** la boîte *NON* et la boîte *Say* au début de la séquence (*onStart*)

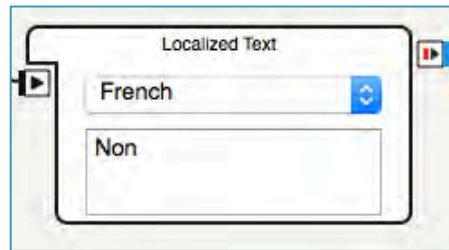
Double-clique sur la boîte *Say* pour accéder aux **détails** :



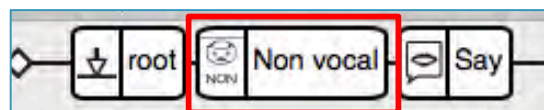
Tu vas quitter ton espace de travail secondaire (intérieur de la boîte *Non vocal*) pour aller dans un **troisième espace de travail** (intérieur de la boîte *Say*):



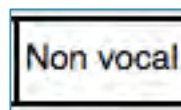
Dans la fenêtre *Localized Text*, choisis la langue *French* (français) en utilisant le menu déroulant :



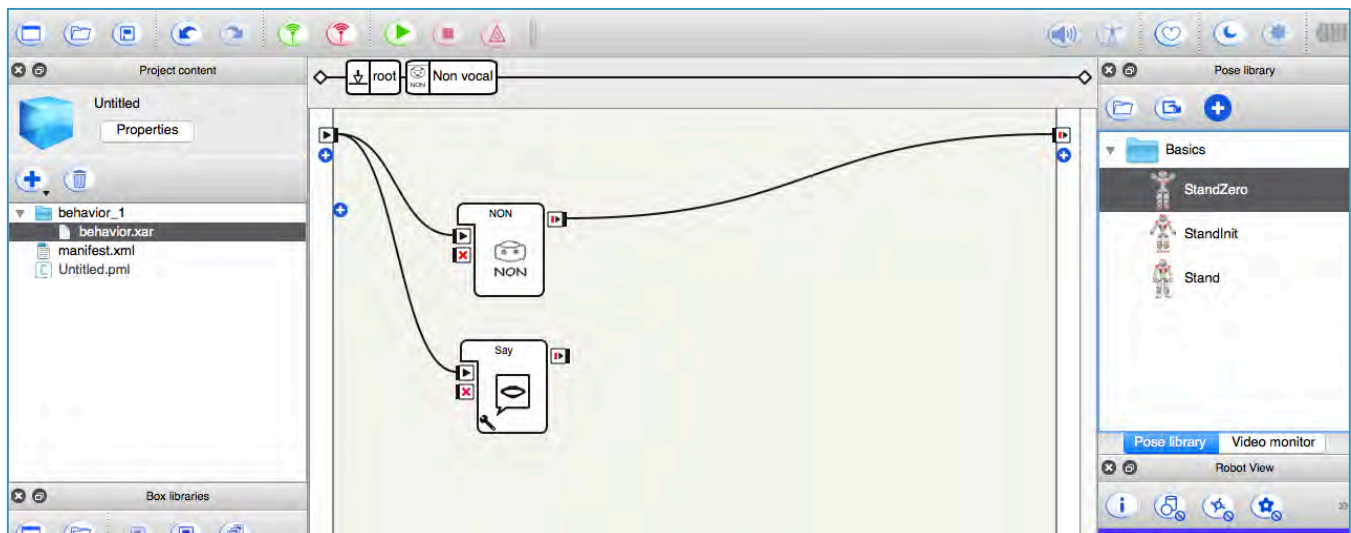
Tu peux ensuite **écrire le message** qui sera prononcé par ton robot NAO : ici, on veut qu'il dise «non».



Quand tu as terminé, **appuie sur** (l'intérieur de la boîte *Non vocal*).

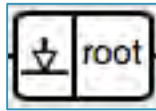


pour revenir dans l'espace de travail secondaire

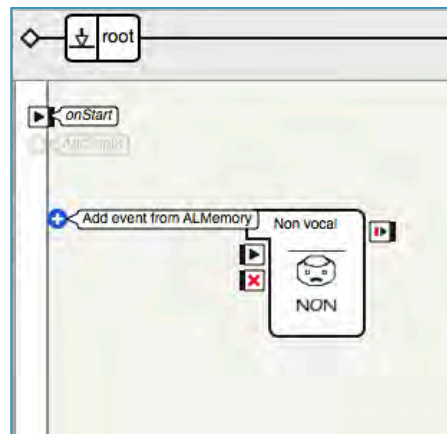


Relie ta boîte *NON* à la fin de ta séquence (*onStop*).

Appuie sur

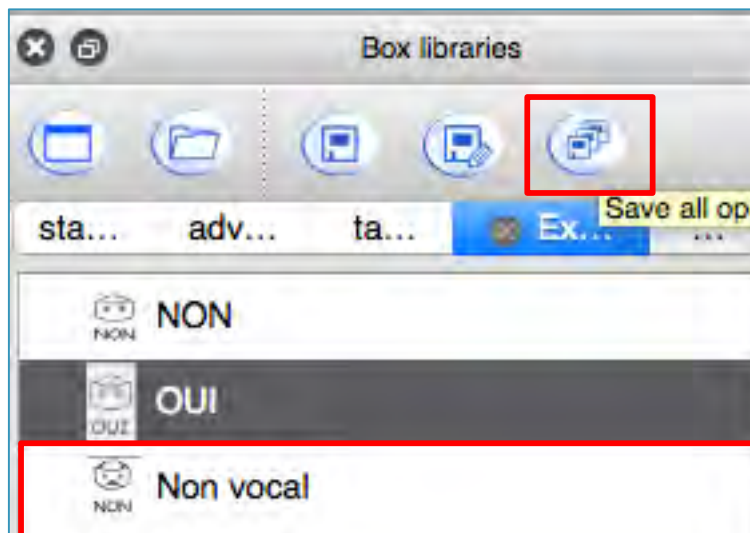


pour retourner à ton espace de travail principal :



ATTENTION

N'oublie pas d'enregistrer ta boîte dans la librairie, sous l'onglet que tu as créé dans la *Partie 1*.





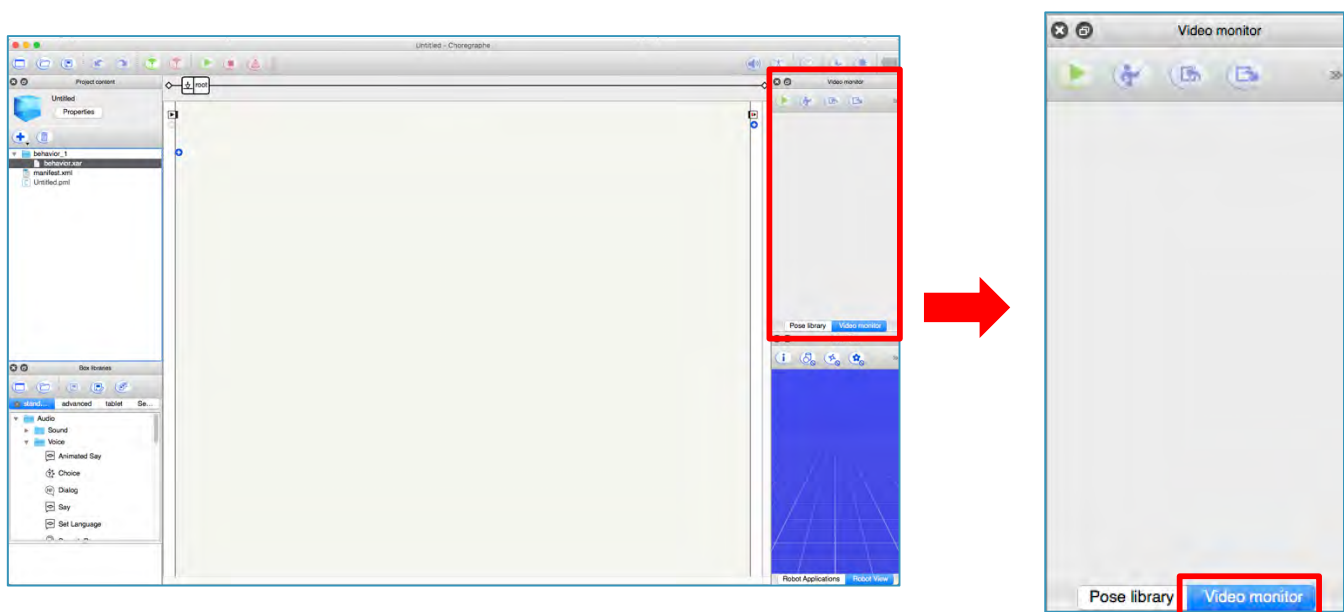
COMMENT UTILISER LA RECONNAISSANCE VISUELLE

SOURCE : ALDEBARAN DOCUMENTATION, CHOREGRAPHE TUTORIALS (TRADUCTION LIBRE)

POUR CRÉER UNE BASE DE DONNÉES DE RECONNAISSANCE

Tu peux apprendre à NAO à reconnaître les choses devant lui. En utilisant la fenêtre *Video monitor*, ton robot peut apprendre des images, des objets ou des lieux (salle de classe).

D'abord, assure-toi **que tu es 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.

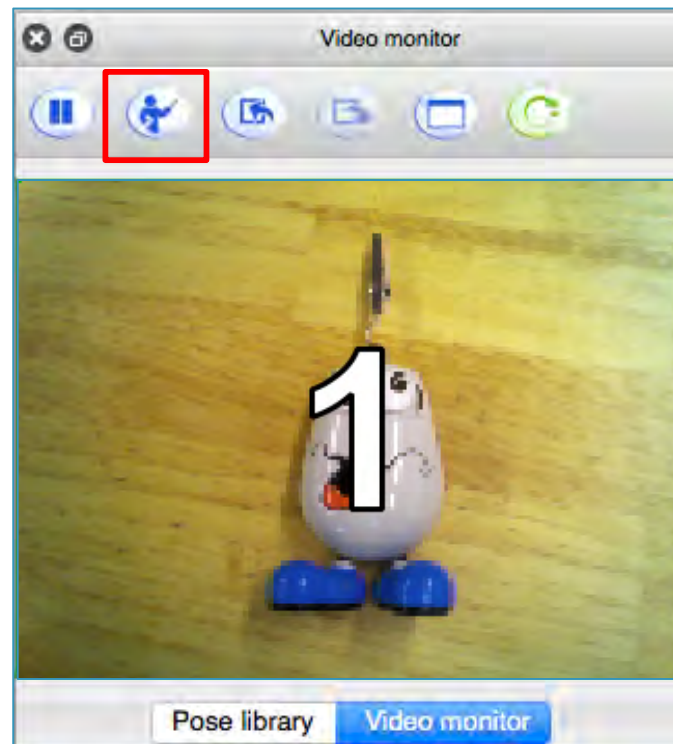
Appuie sur le bouton *Play*



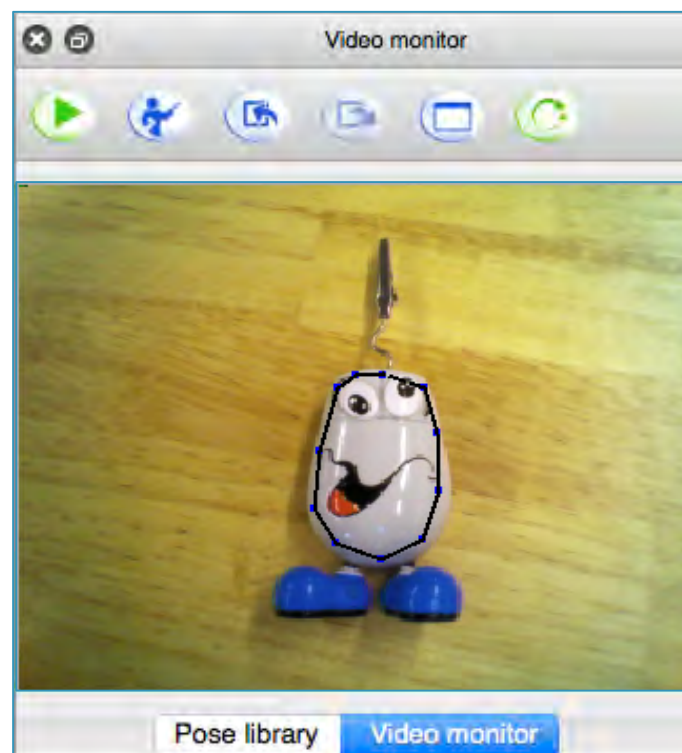
puis sur le bouton *Learn*



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



À zéro, l'image est capturée.



Clique pour dessiner, segment par segment, le contour de l'objet que tu veux faire apprendre à NAO.

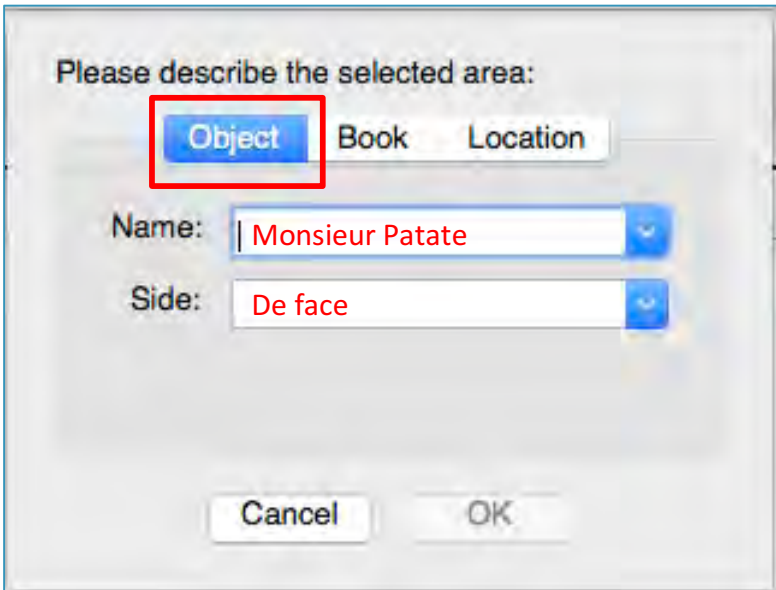
ATTENTION

Si tu veux que NAO reconnaisse une pièce (salle de classe), trace le contour de l'image entière.

Au moment où tu cliqueras à nouveau sur le premier point pour fermer ton contour, **une fenêtre s'ouvre** :

Entre les informations...

Si c'est **un objet**, reste dans l'onglet *Object*.



Please describe the selected area:

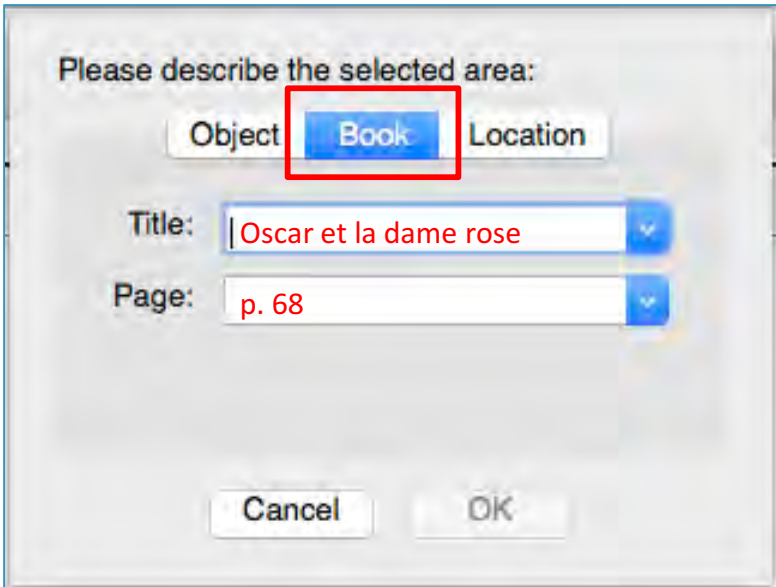
Object Book Location

Name:

Side:

Cancel OK

Si c'est **un livre**, sélectionne l'onglet *Book*.



Please describe the selected area:

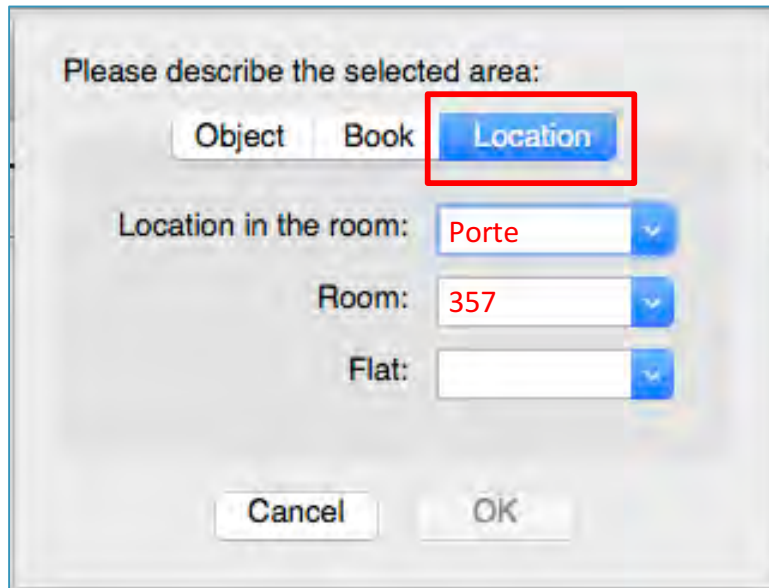
Object **Book** Location

Title:

Page:


Cancel OK

Si c'est un lieu (salle de classe), sélectionne *Location*





Quand tu as terminé d'entrer les informations, **appuie 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.

Si tu veux **faire apprendre d'autres objets** à NAO, appuie sur  et recommence les mêmes étapes.

Quand tu as terminé, tu peux :

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

POUR UTILISER LA RECONNAISSANCE AVEC LE ROBOT

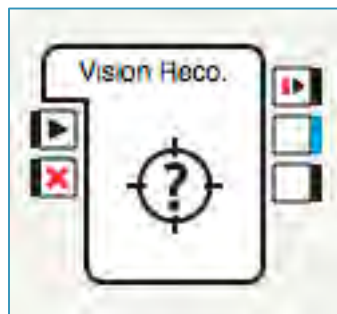
Pour utiliser sur le robot, il faut d'abord **activer** la *vision recognition database* **sur le robot**



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



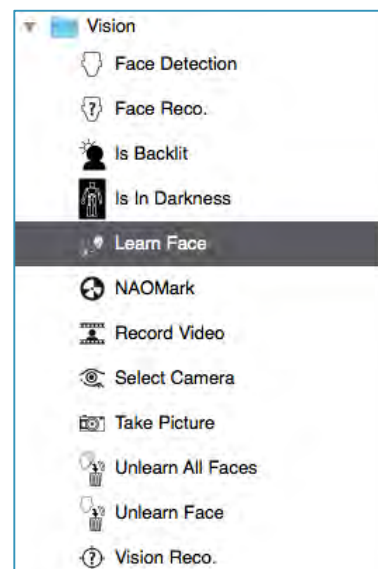
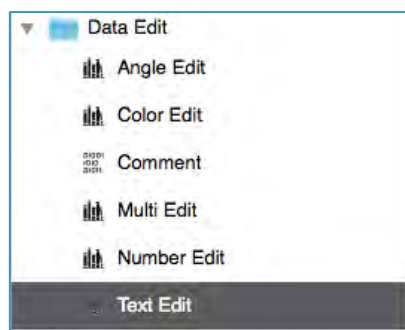
Tu dois ensuite créer un comportement en **utilisant la boîte *Vision Reco*** :



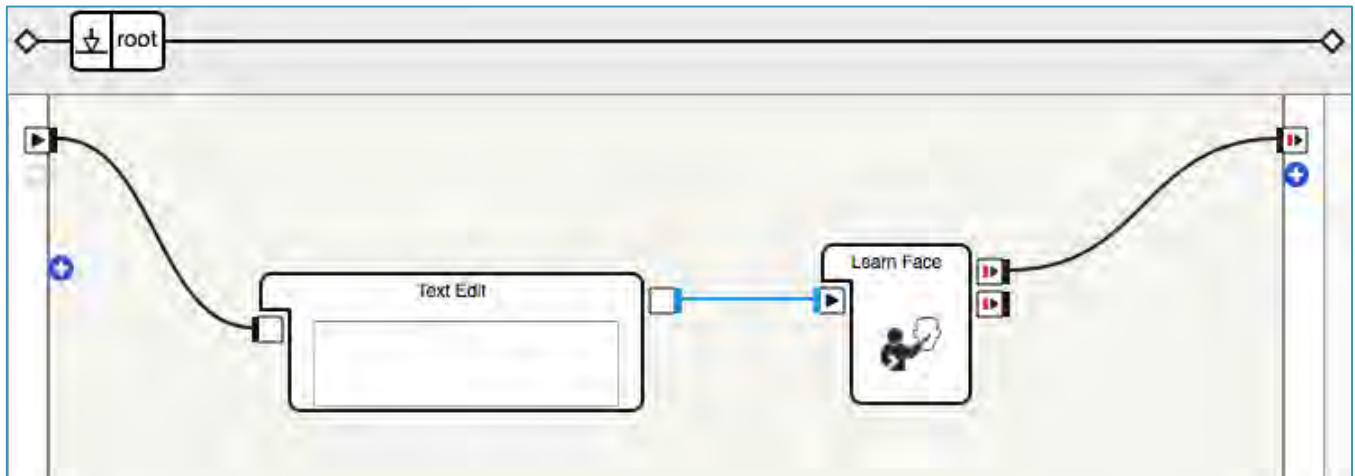
Débute ensuite la séquence pour que NAO puisse reconnaître les objets devant lui.

POUR FAIRE RECONNAÎTRE LES VISAGES

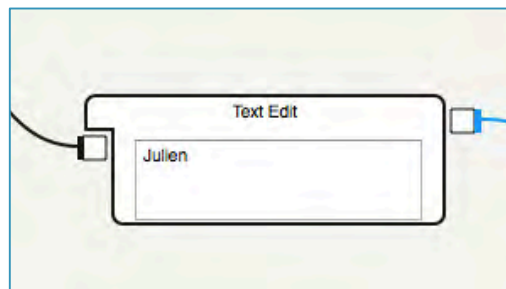
Tu dois d'abord sélectionner les boîtes *Text Edit* et *Learn Face*, puis les déposer dans ton espace de travail en les reliant à *onStart* et *onStop* :



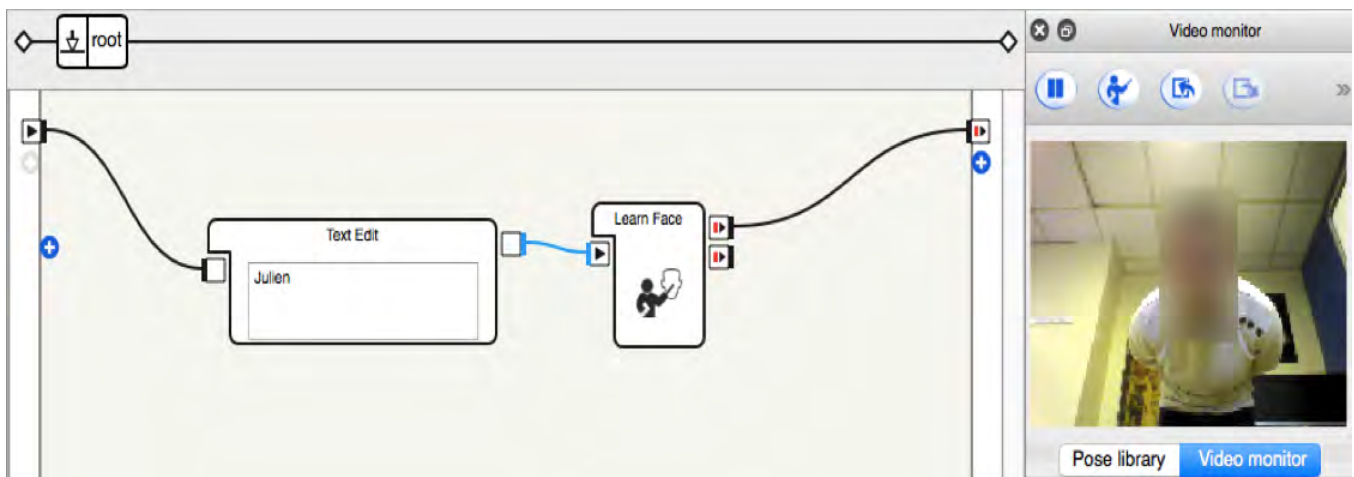
Ton espace de travail doit ressembler à ceci :



Écris **le nom de la personne** dont tu souhaites enregistrer le visage dans la boîte *Text Edit* :

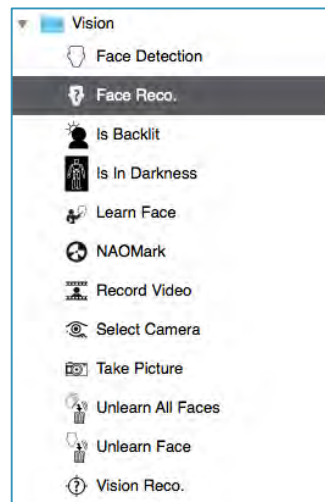


Tu peux ensuite **lancer la séquence** (bouton *Play*) et demander à la personne dont tu souhaites enregistrer le visage de se placer devant les yeux de NAO.



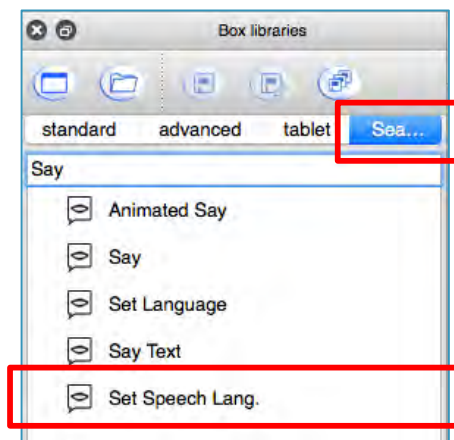
Si NAO a **bien enregistré** le visage, ses yeux deviendront **verts**. Si ses yeux deviennent **rouges**, c'est que quelque chose **n'a pas fonctionné**. Il faut recommencer la manipulation.

Maintenant que le visage est enregistré, tu pourras le faire reconnaître par NAO. Pour ce faire, tu dois utiliser la boîte *Face Reco.*

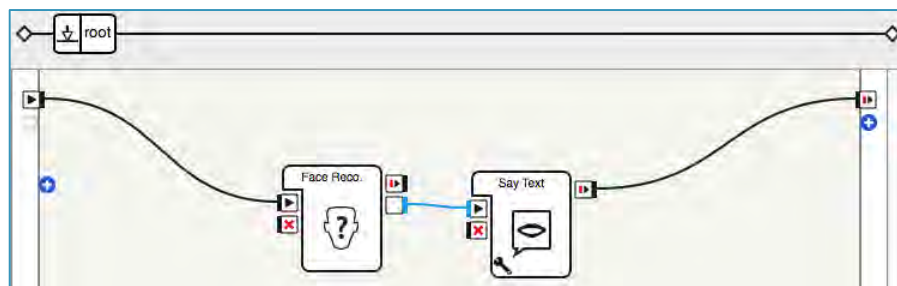


Pour que NAO dise le nom de la personne quand il la voit, tu dois aussi mettre la boîte *Say Text*.

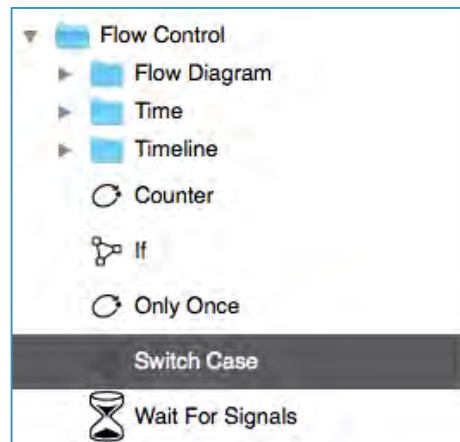
Cette boîte n'est pas dans la liste par défaut : tu dois la chercher dans l'onglet *Search* de la fenêtre *Box Libraries*.



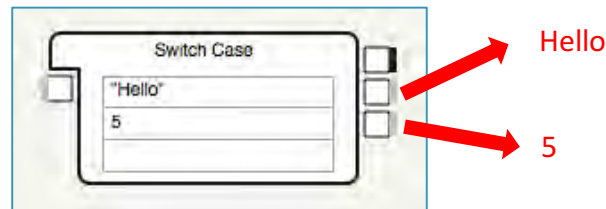
Ton espace de travail devrait ressembler à ceci :



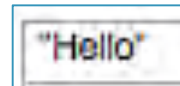
Si tu veux lui faire reconnaître plusieurs visages et faire dire un message personnalisé à NAO, tu peux utiliser la boîte *Switch Case* :



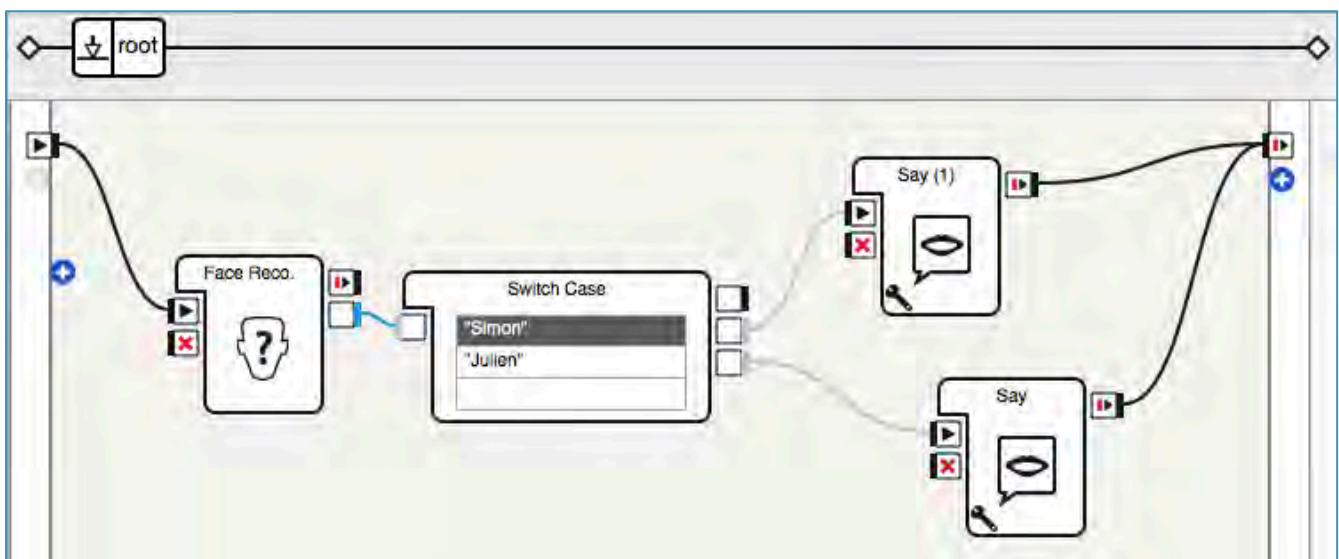
Comme tu peux voir, pour chaque ligne (qui correspond à une information), il y a un *output* :



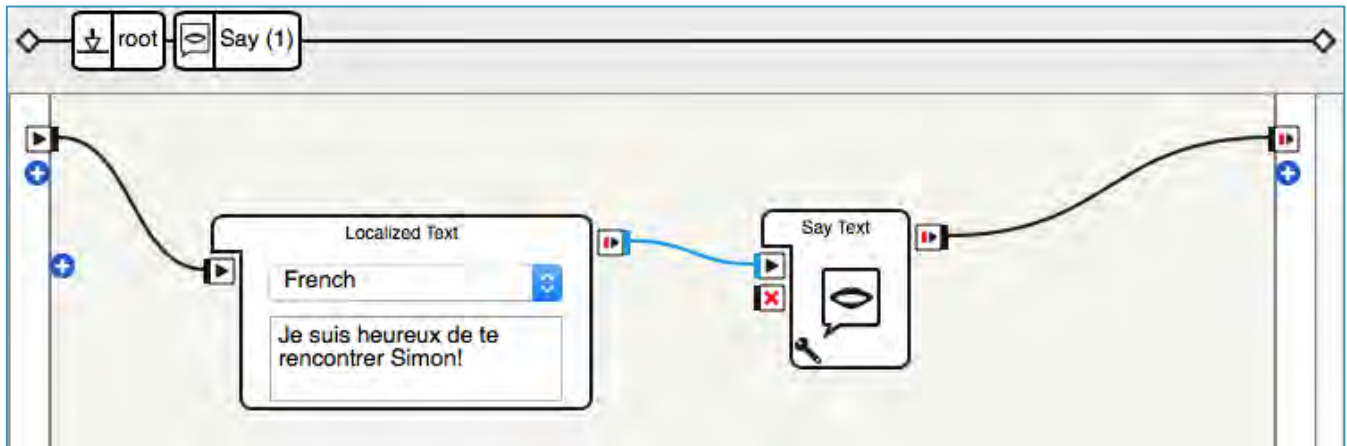
* **IMPORTANT** : les guillemets sont essentiels pour les noms.



Tu peux donc écrire les noms des personnes pour qui tu as enregistré les visages et ajouter des boîtes *Say* pour mettre des messages personnalisés :

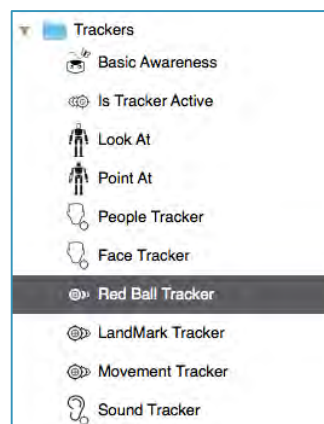


Tu peux ensuite double-cliquer sur les boîtes *Say* pour entrer le message que tu veux faire dire à NAO :

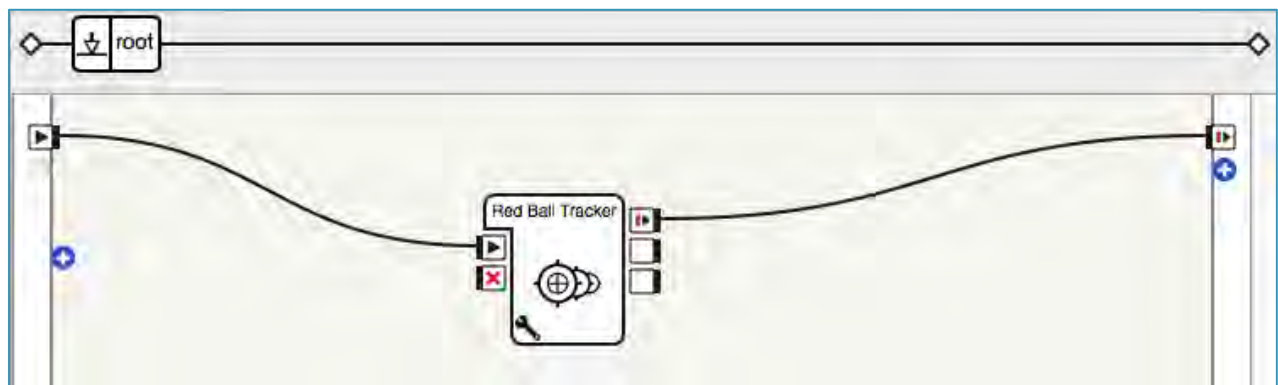


COMMENT UTILISER LA BALLE ROUGE

NAO sait déjà reconnaître la balle rouge : tu n'as donc pas besoin de lui faire apprendre cet objet. Il faut utiliser la boîte *Red Ball Tracker* :

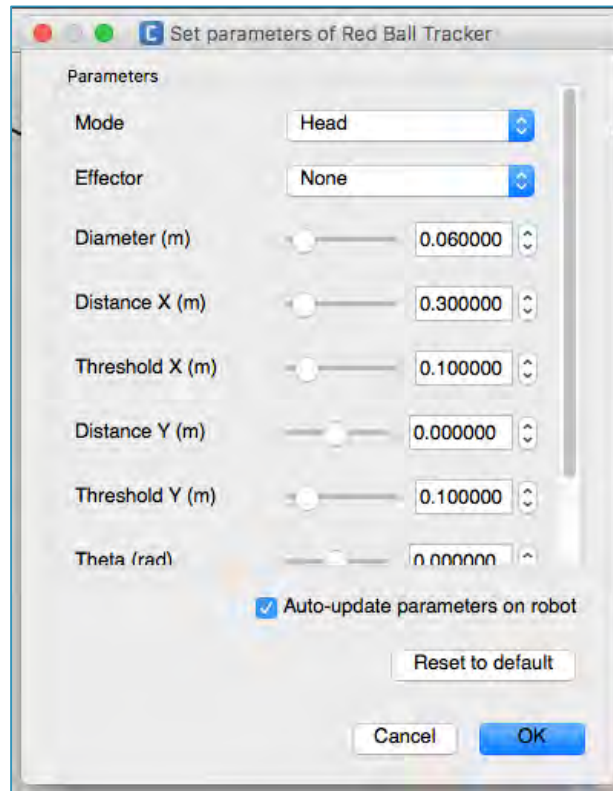


Tu peux la déposer dans ton espace de travail et la lier à *onStart* et *onStop* :

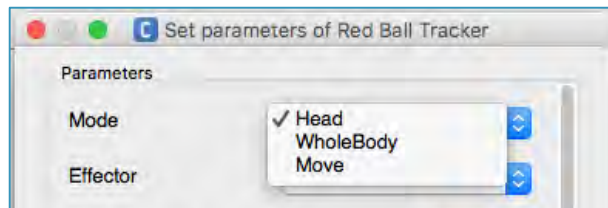


Il y a **plusieurs options** pour cette boîte. Tu peux accéder aux options en appuyant sur l'**outil** dans le coin en bas à gauche de la boîte.

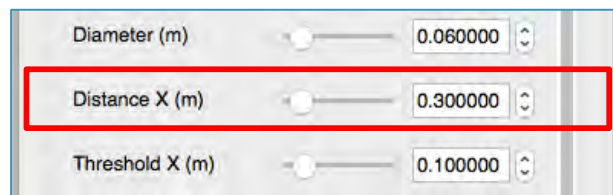
Une **nouvelle fenêtre** va apparaître :



L'option *Mode* te permet de choisir parmi 3 fonctions :



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



Le paramètre **Distance X (m)** te permet de déterminer à **quelle distance de la balle il doit s'arrêter**.

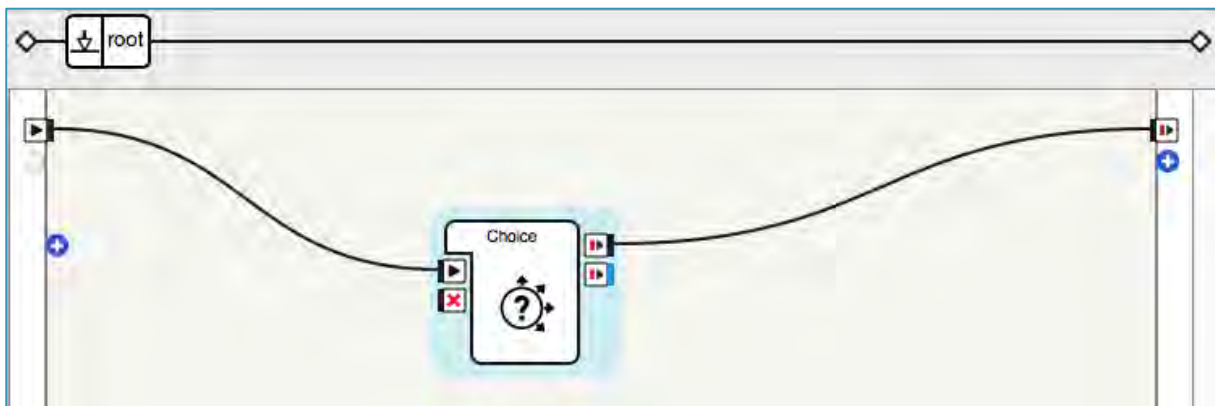
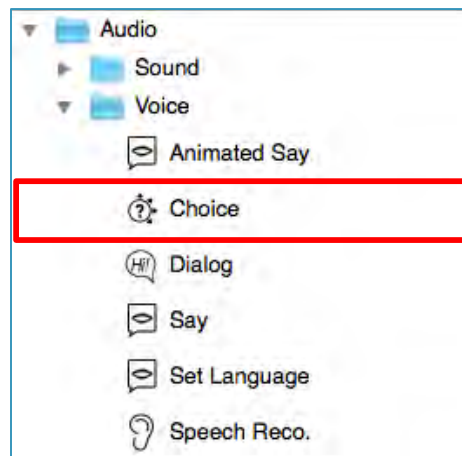


COMMENT UTILISER LA RECONNAISSANCE VOCALE

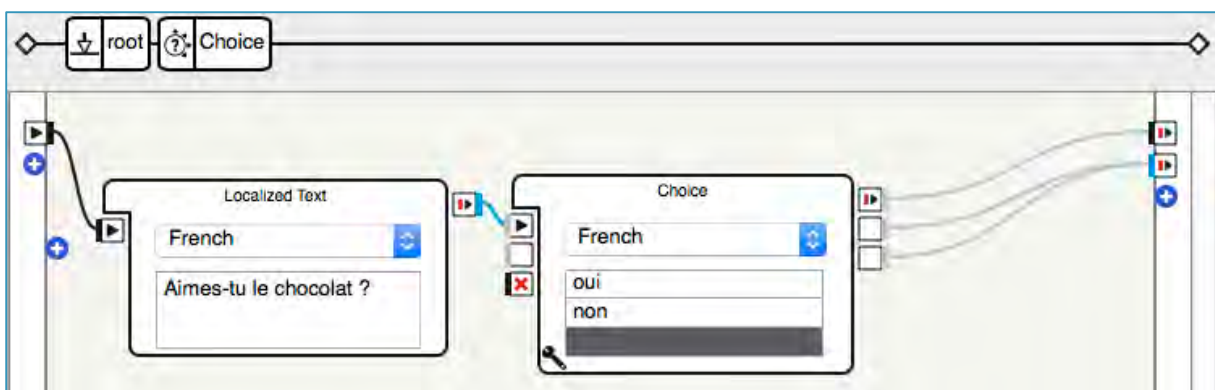
QUESTIONS ET RÉPONSES AVEC NAO

Tu peux utiliser la reconnaissance vocale de NAO pour répondre à ses questions et lui permettre de te dire des commentaires selon tes réponses.

Pour faire poser une question à NAO, dépose la boîte *Choice* dans ton espace de travail :



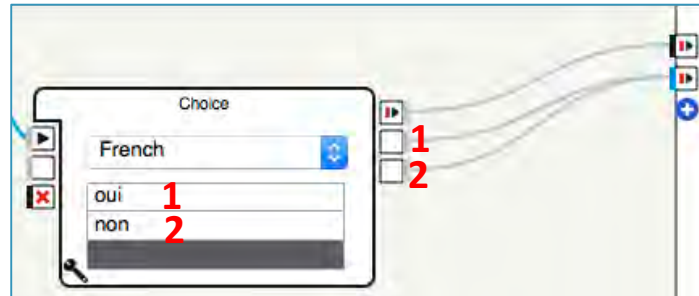
Double-clique sur la boîte *Choice*.



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

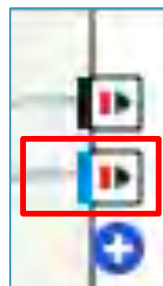
Dans la boîte *Localized Text*, écris **la question** que NAO doit poser. N'oublie pas de choisir l'**option French** pour indiquer que ta **question est en français**.

Dans la boîte *Choice*, écris les réponses que NAO va pouvoir reconnaître.



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

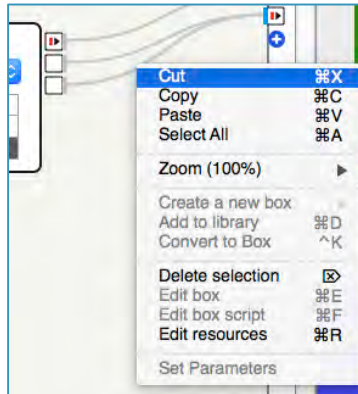
Pour faire une réponse personnalisée, tu dois **supprimer le lien** entre l'*output* et le *onStopped output*, le petit carré avec la ligne bleue :



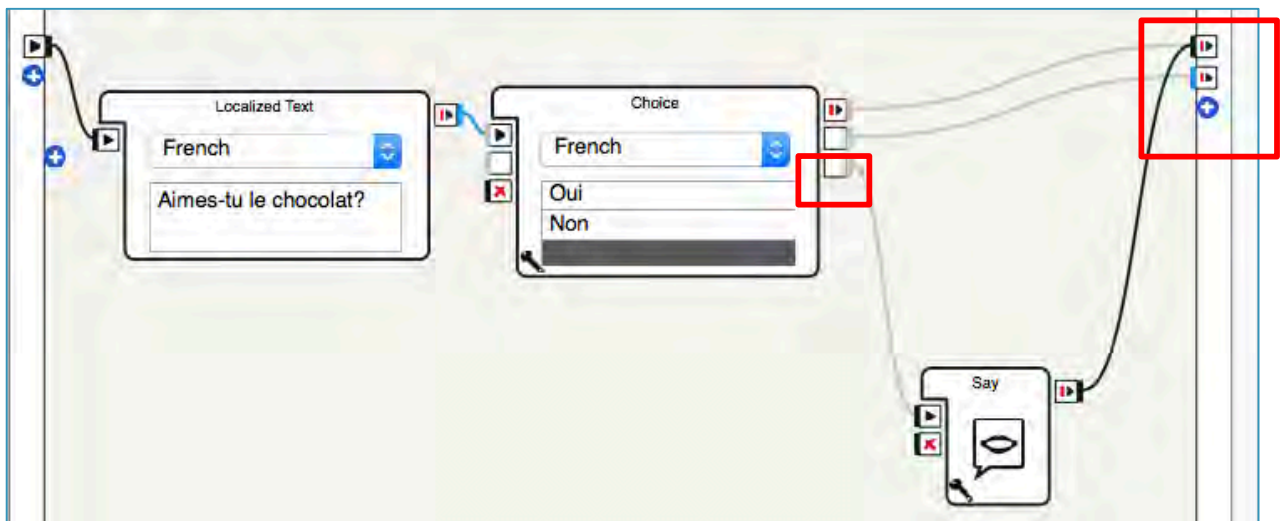
Pour supprimer le lien, sélectionne-le (il va devenir bleu) :



Tu peux ensuite faire un clique-droit avec ta souris et sélectionner l'option *Cut* :

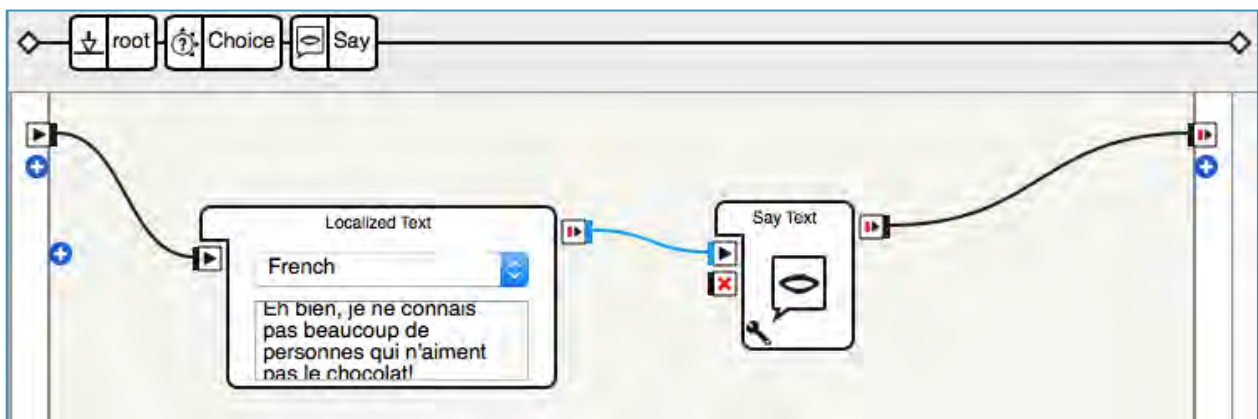


Maintenant que le lien est supprimé, tu peux **ajouter une boîte Say** et la relier comme ci-dessous :



* IMPORTANT : remarque 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.

Double-clique sur la boîte *Say* pour indiquer ce que NAO doit dire si on répond «Non» :



Tu peux **faire la même opération** pour la réponse «Oui».



LE LOGICIEL *CHOREGRAPHE*

PARTIE 3 : *PYTHON*

NIVEAU AVANCÉ

POUR UTILISER *PYTHON*

SOURCE : ALDEBARAN DOCUMENTATION, CHOREGRAPHE TUTORIALS (TRADUCTION LIBRE)



QU'EST-CE QUE *PYTHON* ?

Python est une fonction du logiciel *Choregraphe* qui te permet de créer ou modifier les comportements du robot NAO en utilisant des **lignes de codes**.

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 en modifiant le codage.

LA LOGIQUE DERRIÈRE LE CODAGE

Le langage de codage de programmation dans *Choregraphe* est *Python*. Chaque boîte est **un module**. En résumé, un module est comme **une application**, complètement **indépendante**, qui peut fonctionner seule et **communiquer facilement** avec les autres modules.

La création de boîtes se fait en 3 étapes :

1. Instanciation de la boîte

Je vais réserver un espace dans la mémoire du robot pour la boîte que je vais créer

2. Enregistrement de la boîte en tant que module (pour pouvoir être liée avec d'autres boîtes)

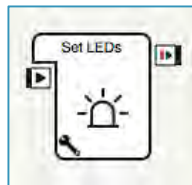
Je dis au logiciel que la boîte est un module, qu'elle pourra communiquer avec les autres

3. Initialisation de la boîte

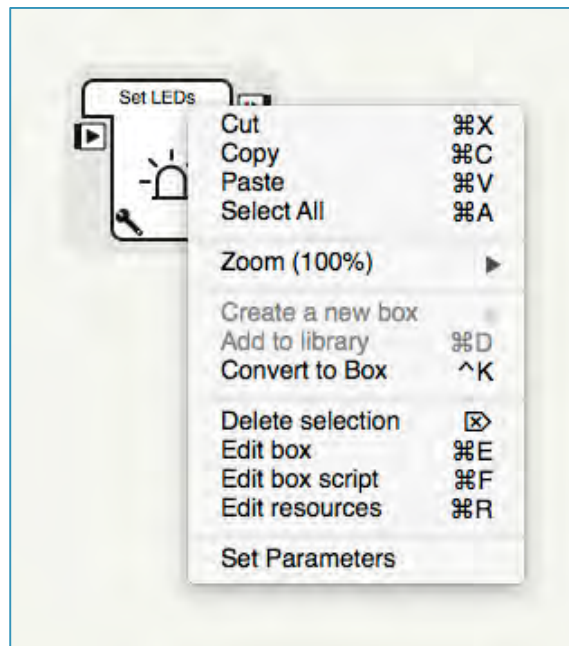
Les deux premières étapes sont automatiques : tu ne peux donc pas changer le codage. Mais tu peux faire plusieurs modifications à la troisième étape.

DESCRIPTION DU CODAGE D'UNE BOÎTE

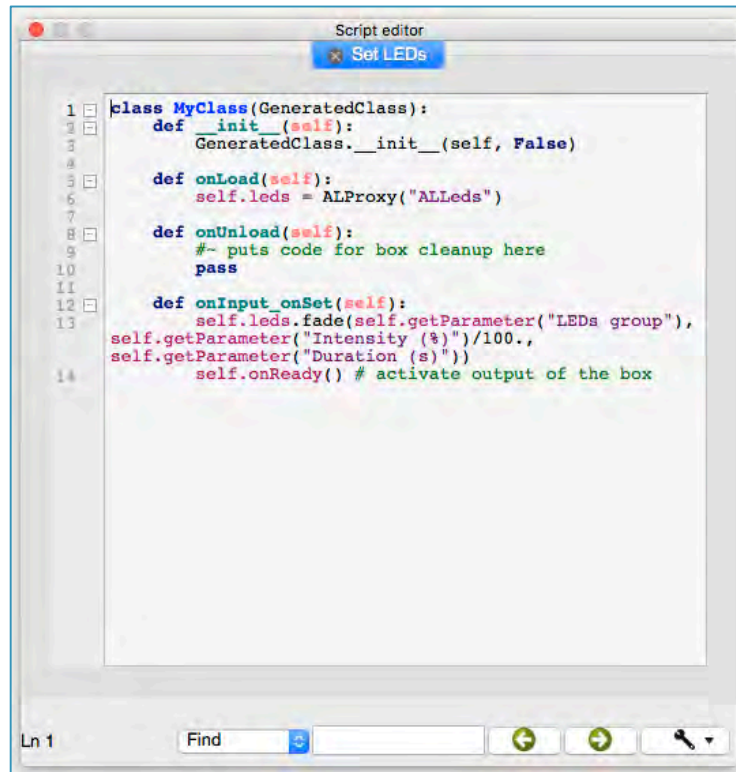
Prenons en exemple la boîte *Set LEDs*.



Pour **voir le codage** d'une boîte, **clique-droit** sur une boîte et appuie sur *Edit box script* (ou CMD+F) :



La fenêtre *Script editor* va apparaître :

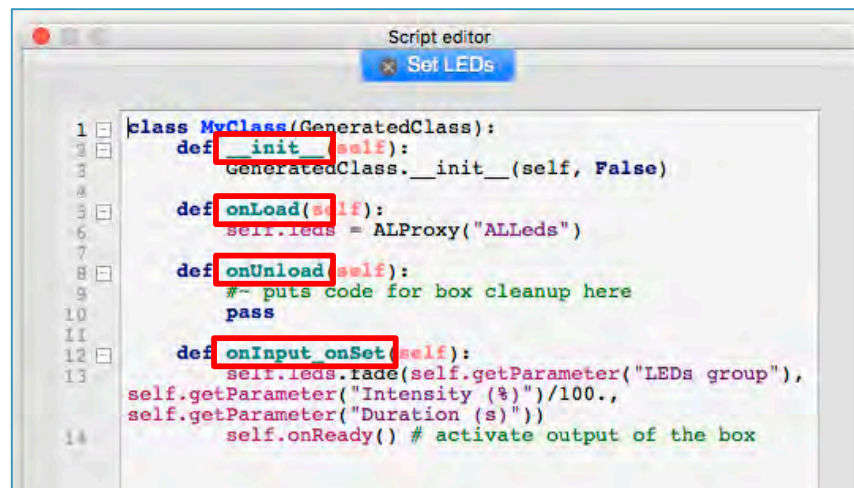


The screenshot shows a 'Script editor' window with a tab labeled 'Set LEDs'. The code defines a class 'MyClass' that inherits from 'GeneratedClass'. It includes four methods: 'init', 'onLoad', 'onUnload', and 'onInput_onSet'. The 'onInput_onSet' method calls 'self.leds.fade' and 'self.onReady'.

```
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
```

Quand tu écris le codage, tu dois te souvenir que tu es en train de définir **les méthodes** d'une classe de module.

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



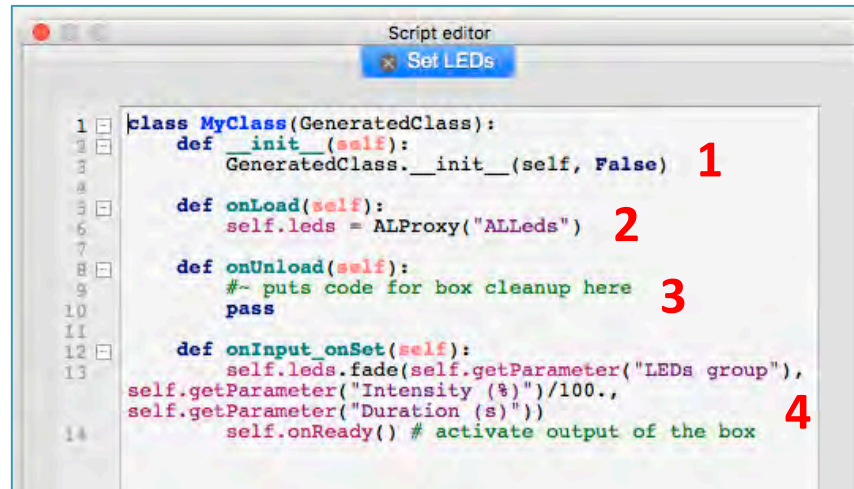
This screenshot is identical to the previous one, but the method names 'init', 'onLoad', 'onUnload', and 'onInput_onSet' are each enclosed in a red rectangular box to highlight them.

1. Initialize
2. onLoad
3. onUnload
4. onInput_onSet

COMMENT FONCTIONNE LE CODAGE

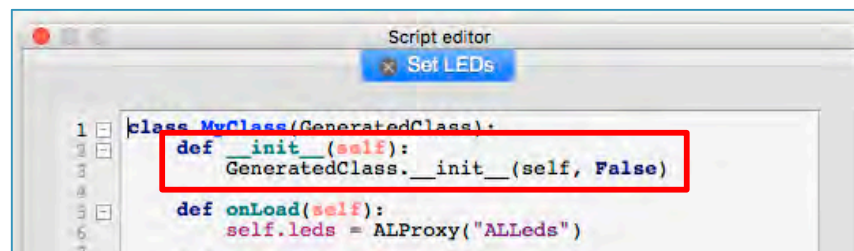
On continue avec l'exemple de la boîte *Set LEDs*...

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 3
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)")) 4
16        self.onReady() # activate output of the box
```

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*). Après cela, la boîte est un **module** qui est exécuté dans NAOqi, dans le robot.



ATTENTION

Tu ne dois **jamais modifier les 3 premières lignes** de codage d'une boîte puisqu'elles sont obligatoires !

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* de toutes les boîtes du comportement.

```
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
```

Quand un diagramme de flux est en chargement, cette méthode est exécutée sur chaque boîte d'un même niveau avant qu'un IO puisse être simulé.

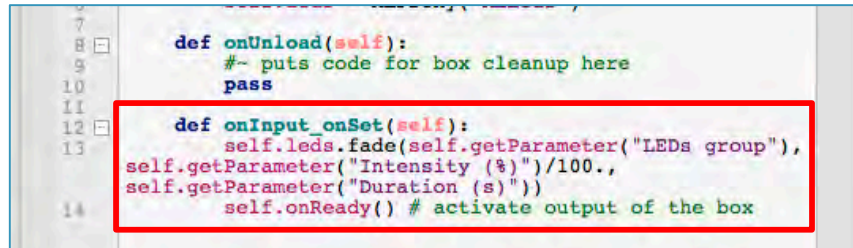
3. La **méthode onUnload** : cette méthode est exécutée quand le diagramme de flux de la boîte est en déchargement. Lors du déchargement, cette méthode est exécutée sur toutes les boîtes d'un même niveau.

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

Après le déchargement, les boîtes sont désactivées et ne peuvent plus recevoir d'événement ou d'informations (*inputs*). N.B. : la méthode arrête habituellement tout ce qui est *actif* dans le codage, d'où ce qu'on anticipe de l'*input onStop*. Voilà pourquoi ce dernier exécute *onUnload* par défaut.

4. La **méthode onInput_onSet** : cette méthode est exécutée quand le *onStart* est simulé. 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)"))
```



```
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
```

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 la méthode **onStart**, *self.onStopped()* simule le *output onStopped* de la boîte *Switch LEDs*.

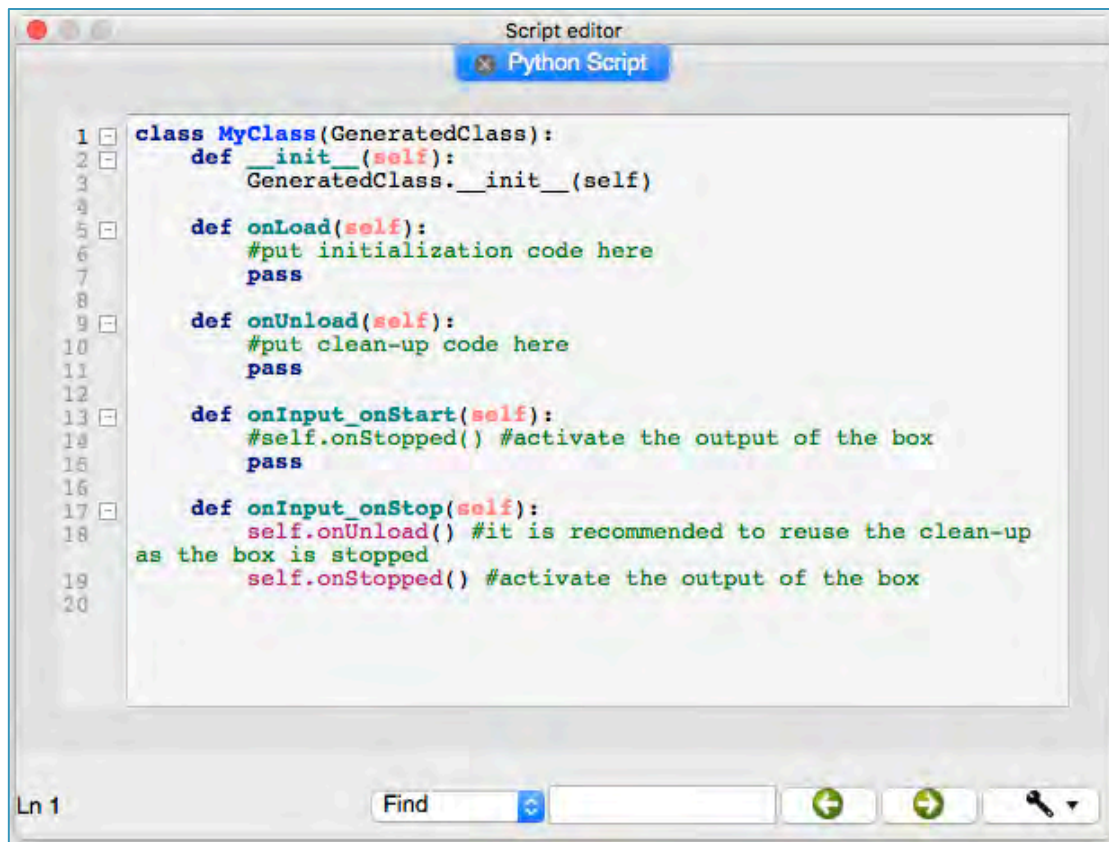
POUR FAIRE PARLER NAO AVEC PYTHON

Tu dois d'abord ajouter une boîte *Python Script* sur ton espace de travail :



Tu peux maintenant double-cliquer sur la boîte pour accéder au codage de la boîte.

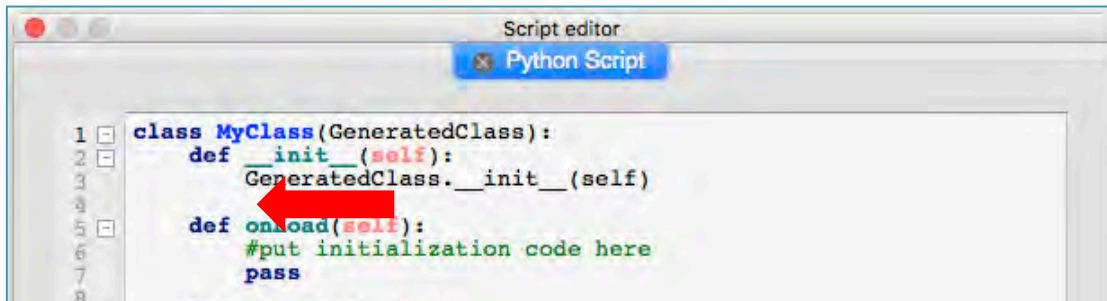
Cette fenêtre va apparaître :

A screenshot of a 'Script editor' window titled 'Python Script'. The window contains a code editor with the following Python code:

```
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
```

The code is color-coded: keywords in blue, comments in green, and strings in red. The window has a standard macOS-style title bar with red, yellow, and green buttons. At the bottom, there is a 'Find' search bar and several navigation icons.

Tu dois ensuite identifier quel est le module que tu souhaites utiliser avec cette boîte.

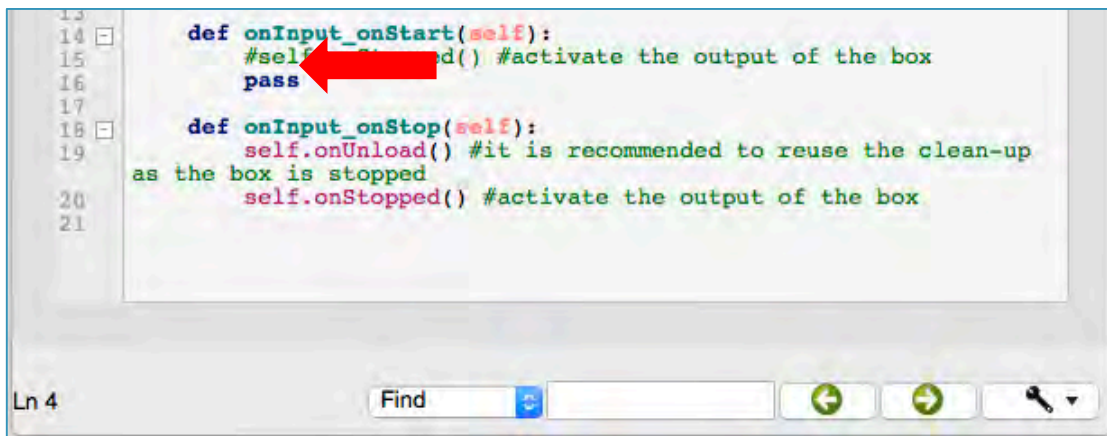


```
1 class MyClass(GeneratedClass):
2     def __init__(self):
3         GeneratedClass.__init__(self)
4
5     def onload(self):
6         #put initialization code here
7         pass
```

Tu dois ajouter une ligne de code après la 3^e ligne :

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

Tu dois maintenant ajouter une ligne de code après la 15^e ligne :



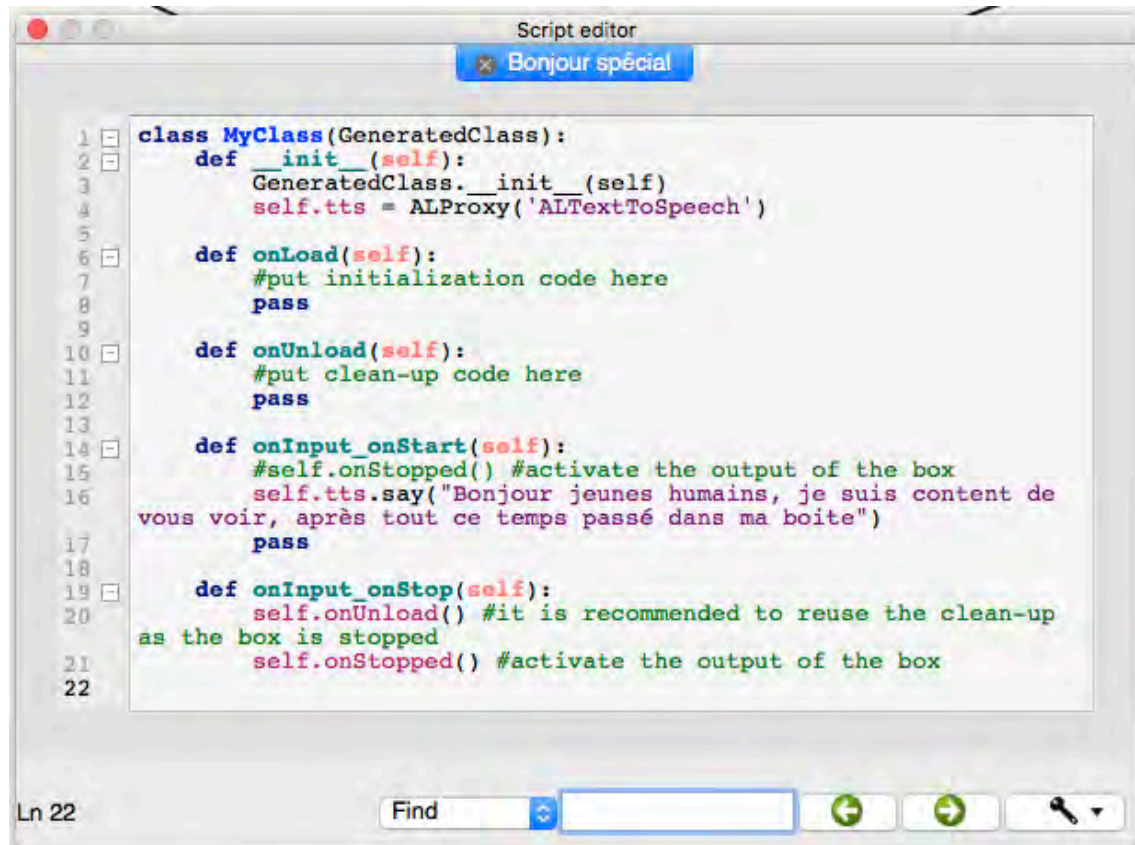
```
13
14 def onInput_onStart(self):
15     #self.onStopped() #activate the output of the box
16     pass
17
18 def onInput_onStop(self):
19     self.onUnload() #it is recommended to reuse the clean-up
20     as the box is stopped
21     self.onStopped() #activate the output of the box
```

Voici le texte à ajouter :

```
def onInput_onStart(self):
    #self.onStopped() #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, NAO 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 ta 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
vous voir, après tout ce temps passé dans ma boîte")
17        pass
18
19    def onInput_onStop(self):
20        self.onUnload() #it is recommended to reuse the clean-up
as the box is stopped
21        self.onStopped() #activate the output of the box
22
```

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

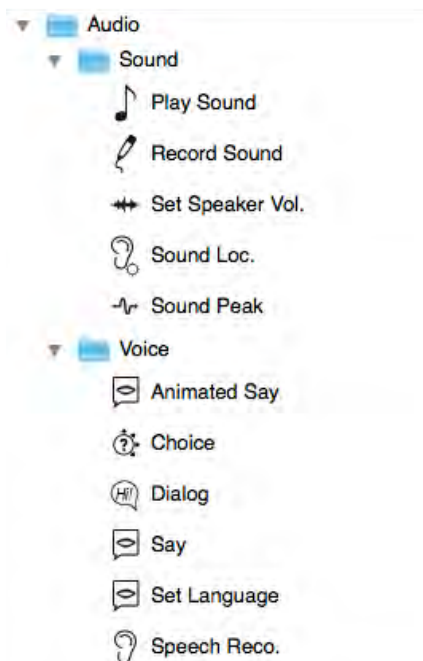
```
( 'ALTextToSpeech' )
```

et


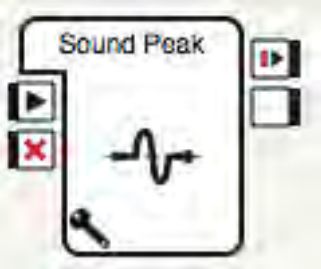
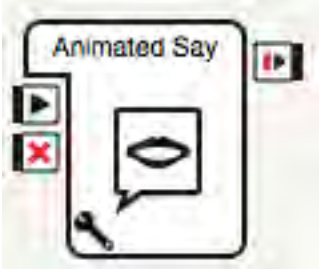
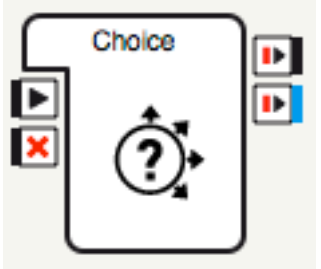

```
("Bonjour jeunes hu
```

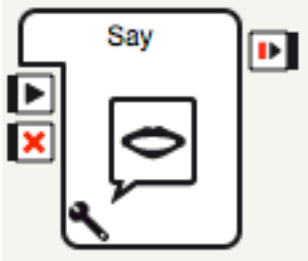
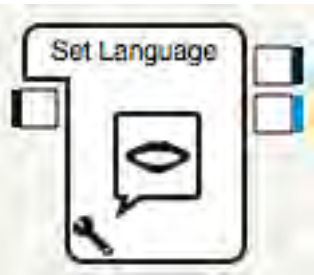

```
ssé dans ma boîte")
```


AUDIO

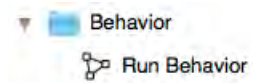


	<h3>Jouer un son</h3>	<p>Permet de jouer un fichier sonore. Il faut sélectionner le fichier dans les paramètres.</p> <p>Le format peut être <i>.wav</i>, <i>.ogg</i> ou <i>.mp3</i> (le format <i>.wav</i> permet une lecture plus rapide).</p>
	<h3>Enregistrer un son</h3>	<p>Permet d'enregistrer un son sur le robot à partir du microphone choisi.</p> <p>L'<i>output</i> est stimulé à la fin de l'enregistrement.</p>
	<h3>Ajuster le volume des haut-parleurs</h3>	<p>Définir le volume du son entre 0 (sourdine) et 100 (maximum).</p>

	<h2>Localisation du son</h2>	<p>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.</p>
	<h2>Détecteur de pic sonore</h2>	<p>Détecte les pics sonores. Chaque fois qu'il y a un pic sonore, l'<i>output</i> est stimulé.</p> <p>Vous pouvez ajuster la sensibilité de détection dans les paramètres.</p>
	<h2>Discours animé (mouvements)</h2>	<p>Parler de façon animée.</p> <p>Vous devez ouvrir la boîte <i>Animated Say</i> pour entrer le texte à dire.</p>
	<h2>Choix de réponses</h2>	<p>Le choix est basé sur la reconnaissance vocale : on s'attend à un certain nombre de réponses prédéfinies.</p> <p>La boîte inclut de l'aide interactive pendant le comportement : simplement appuyer sur les capteurs sensoriels de la tête.</p>
	<h2>Dialogue</h2>	<p>Un exemple de la mise en œuvre d'un dialogue plurilingue.</p>

	<p>Dire...</p>	<p>Dire des lignes de textes écrites dans la boîte.</p>
	<p>Programmer la langue</p>	<p>Permet de sélectionner la langue qui sera comprise et parlée par NAO. Toutes les boîtes suivantes relatives au langage vont utiliser les langues programmées ici.</p>
	<p>Reconnaissance vocale</p>	<p>Reconnaît des mots à partir d'une liste écrite dans les paramètres de la boîte.</p>

BEHAVIOR (COMPORTEMENT)






Démarrer un comportement


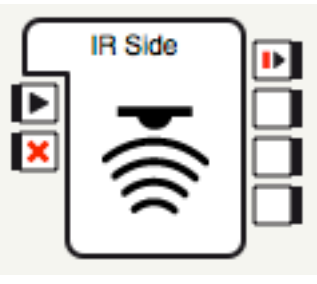

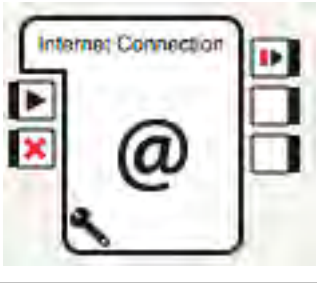
Permet de gérer un comportement.

Début le comportement défini en tant qu'*input* lorsqu'*onStart* est stimulé, et l'arrête lorsque *onStop* est stimulé.

COMMUNICATION



	<h2>Recevoir un courriel</h2>	<p>Permet de recevoir des courriels à partir d'un serveur, qui doit être programmé dans les paramètres.</p> <p>Note : s'assurer que POP est activé sur le serveur.</p> <p>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.</p>
	<h2>Envoyer un courriel</h2>	<p>Permet d'envoyer des courriels. Définir tous les paramètres de ce courriel avant de l'envoyer.</p>
	<h2>Receveur de signal infrarouge</h2>	<p>Fait fonctionner NAO comme le receveur d'une télécommande.</p> <p>Vous devez définir les paramètres de la télécommande (il faut une clé : voir la page Web de NAO).</p>

	<p>Émetteur de signal infrarouge</p>	<p>Fait fonctionner NAO comme une télécommande.</p> <p>Il faut définir les paramètres la clé (voir la page Web de NAO).</p>
	<p>Côté du receveur d'infrarouge (gauche/droite)</p>	<p>Si NAO reçoit un signal infrarouge principalement dans l'œil droit, il va activer l'<i>output</i> «œil droit». Même principe pour l'œil gauche.</p>
	<p>Vérifier la connexion à <i>Choregraphe</i></p>	<p>Vérifier si le robot est connecté à <i>Choregraphe</i> et avertit lors d'une connexion ou déconnexion.</p>
	<p>Vérifier la connexion à Internet</p>	<p>Détecte une connexion (ou l'absence d'une connexion) à Internet.</p>

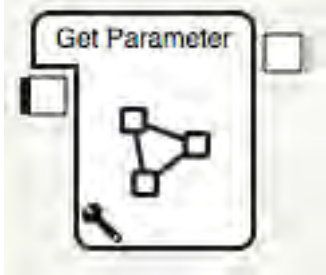


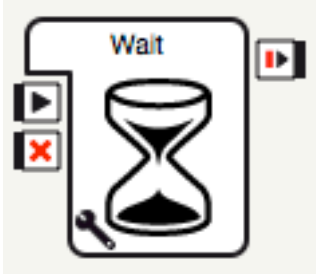

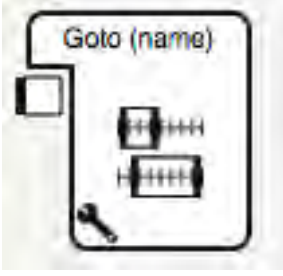
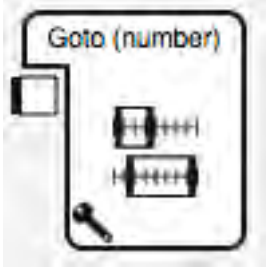
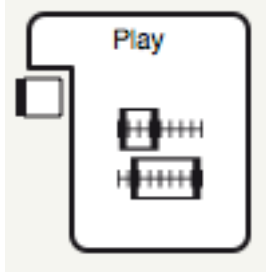
	<h2>Définir un angle</h2>	<p>Envoie un angle. L'angle est converti en radians, puis est envoyé à travers l'<i>output</i>.</p>
	<h2>Définir une couleur</h2>	<p>Envoie une couleur encodée en valeurs de RVB.</p>
	<h2>Commentaire</h2>	<p>Permet de commenter le comportement.</p> <p>Cette boîte n'est pas fonctionnelle et n'a aucun effet sur le comportement.</p>
	<h2>Définir plusieurs valeurs</h2>	<p>Envoie la valeur qui se trouve devant l'<i>input</i> stimulé (peut être tous les types de langages de <i>Python</i> : string, int, float, array, etc.).</p> <p>Vous pouvez définir une valeur en double-cliquant sur la ligne. Pour insérer une ligne : clique-droit + <i>insert row</i>.</p>
	<h2>Définir un nombre</h2>	<p>Envoie le nombre défini lorsque l'<i>input</i> est stimulé.</p>

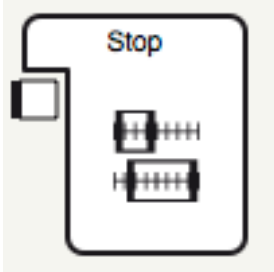
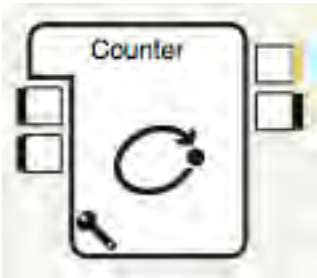
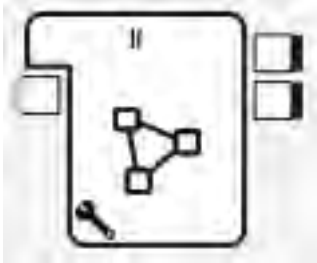
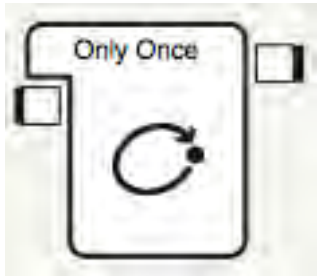
	<h2>Définir un texte</h2>	<p>Envoie le texte défini lorsque l'<i>input</i> est stimulé.</p>
---	---------------------------	---

FLOW CONTROL (CONTRÔLE DE DÉROULEMENT)



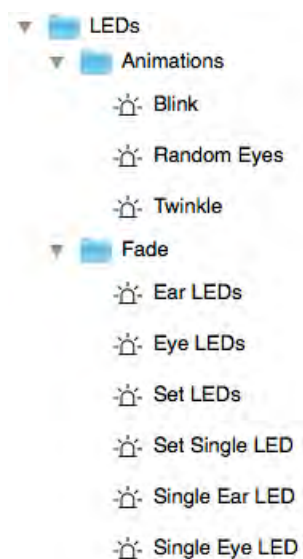
	<h2>Obtenir le paramètre</h2>	<p>Retourne la valeur du paramètre de la boîte parente. Établir le nom du paramètre qui devra être retourné dans les paramètres de cette boîte.</p>
	<h2>Minuteur</h2>	<p>Stimule le <i>output</i> de façon répétitive avec un intervalle spécifié.</p> <p>Peut être arrêté à tout moment. Stimuler le <i>input</i> à nouveau remet le minuteur à zéro.</p>

	<p>Attente</p>	<p>Attendre un moment avant d'envoyer le signal à l'<i>output</i>.</p> <p>Peut être arrêté à tout moment. Stimuler le <i>input</i> à nouveau avant que l'<i>output</i> soit activé recommence la période d'attente.</p>
	<p>Délai</p>	<p>Attendre un moment avant d'activer l'<i>output</i>.</p> <p>Peut être arrêté à tout moment. Plusieurs <i>inputs</i> vont déclencher plusieurs <i>outputs</i>.</p>
	<p>Aller à (nom)</p>	<p>Sauter au nom de l'image (<i>frame</i>) programmé dans les paramètres.</p> <p>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).</p>
	<p>Aller à (nombre)</p>	<p>Sauter au nombre de l'image (<i>frame</i>) programmé dans les paramètres.</p>
	<p>Jouer</p>	<p>Jouer la ligne du temps parente depuis l'index d'image actuel.</p> <p>Ne fait rien si la ligne du temps est déjà en train de jouer.</p>



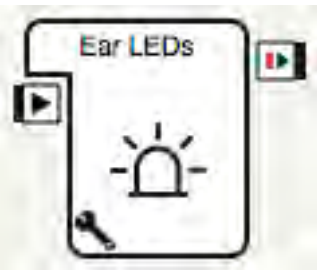
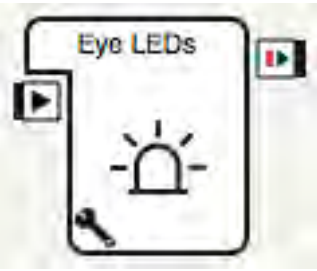
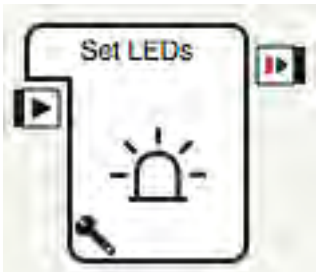
	<p>Arrêt</p>	<p>Arrête la ligne du temps parente à l'index d'image actuel.</p>
	<p>Compteur</p>	<p>Représente un compteur.</p> <p>Chaque fois que l'<i>input onStart</i> est stimulé, la valeur du compteur est envoyée à l'<i>output currentValue</i>, et incrémentée (ou décrémentée) par <i>Step value</i>, 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>
	<p>Si...</p>	<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>
	<p>Seulement une fois</p>	<p>L'<i>output</i> de cette boîte est stimulé seulement la première fois que l'<i>input onSignal</i> est envoyé.</p> <p>Note : Elle est réinitialisée quand le diagramme contenant la boîte est <i>unloaded</i>.</p>

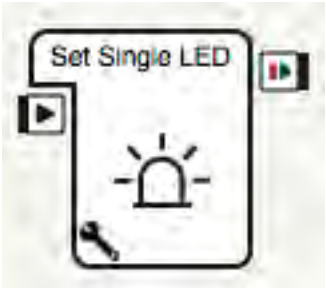

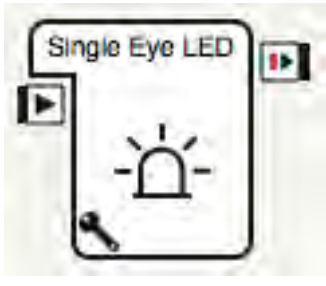
	<h2>Boîtier de commutation</h2>	<p>Vérifie la valeur de l'<i>input</i> et stimule l'<i>output</i> correspondant. S'il n'y a pas d'<i>output</i> correspondant, l'<i>output par défaut</i> est stimulé.</p>
	<h2>Attendre les signaux</h2>	<p>Attend que les deux <i>inputs</i> soient stimulés avant de stimuler l'<i>output</i>.</p>

LEDs (DIODES ÉLECTROLUMINESCENTES)

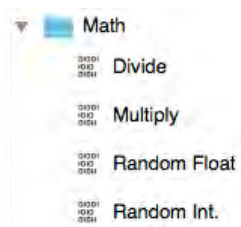


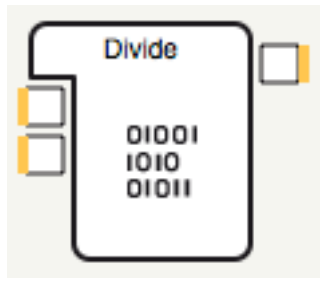
	<h2>Clignotement</h2>	<p>Fait clignoter les yeux de NAO une seule fois.</p>
---	-----------------------	---

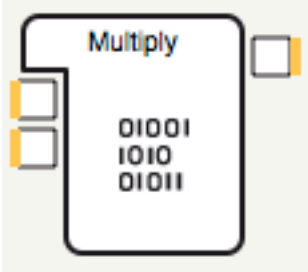

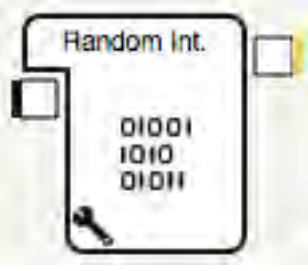
	<h2>Yeux aléatoires</h2>	<p>Fait changer la couleur des yeux de façon aléatoire.</p> <p>Note : c'est une boîte qui ne s'arrête pas. Vous devez stimuler l'<i>input onStop</i> pour l'arrêter.</p>
	<h2>Clignotement continu</h2>	<p>Les DEL sélectionnées clignent pendant une durée définie dans les paramètres.</p>
	<h2>DEL d'oreille</h2>	<p>Défini l'intensité de toutes les DEL d'une ou des deux oreilles.</p>
	<h2>DEL d'oeil</h2>	<p>Défini la couleur des yeux.</p> <p>Vous devez ouvrir la boîte pour définir la couleur.</p>
	<h2>Programmer les DEL</h2>	<p>Défini l'intensité des DEL d'un groupe choisi (dans la paramètres).</p>

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

MATH (MATHÉMATIQUE)






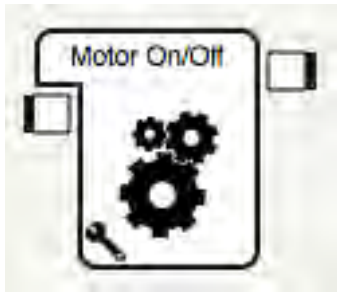
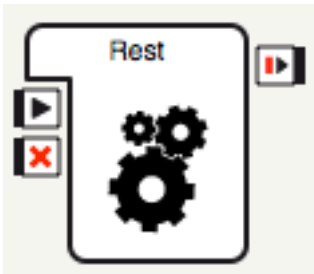
	<p>Diviser</p>	<p>Divise la valeur du premier <i>input</i> par la valeur du deuxième <i>input</i> et envoie le résultat à l'<i>output</i>.</p>
---	-----------------------	---

 <p>The icon shows a block labeled 'Multiply' with two input ports on the left and one output port on the right. Inside the block, the binary code 01001 1010 01011 is displayed.</p>	<h2>Multiplier</h2>	<p>Multiplie la valeur du premier <i>input</i> par la valeur du deuxième <i>input</i> et envoie le résultat à l'<i>output</i>.</p>
 <p>The icon shows a block labeled 'Random Float' with two input ports on the left and one output port on the right. Inside the block, the binary code 01001 1010 01011 is displayed, along with a key icon at the bottom left.</p>	<h2>Nombre en virgule flottante aléatoire</h2>	<p>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).</p> <p>Ex. :</p> $1.3254 = \underbrace{13254}_{\text{mantisse}} \times 10^{\overbrace{-4}^{\text{exposant}}}$
 <p>The icon shows a block labeled 'Random Int.' with two input ports on the left and one output port on the right. Inside the block, the binary code 01001 1010 01011 is displayed, along with a key icon at the bottom left.</p>	<h2>Nombre entier aléatoire</h2>	<p>Génère un nombre entier compris entre un nombre minimal et un nombre maximal, programmés dans les paramètres.</p>

MOTIONS (MOUVEMENTS)



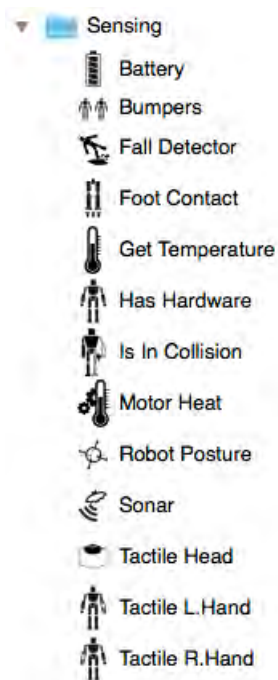
	<h1>Respirer</h1>	<p>Activer ou désactiver l'animation de respiration sur la chaîne désirée.</p>
	<h1>Bonjour</h1>	<p>Animation simple de «Bonjour !».</p>

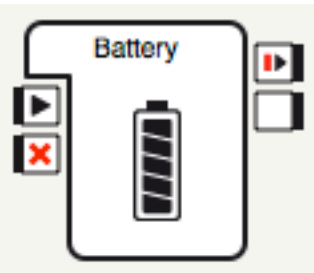
	<p>Essuyer le front</p>	<p>NAO s'essuie le front.</p>
	<p><i>Tai Chi Chuan</i></p>	<p>Danse <i>Tai Chi Chuan</i>.</p>
	<p>Mains</p>	<p>NAO raidit les moteurs d'une ou de ses mains pour qu'il puisse la ou les fermer/ouvrir.</p> <p>Il détend ensuite les moteurs de ses mains.</p>
	<p>Moteurs Marche/Arrêt</p>	<p>Raidit ou détend tous les moteurs.</p>
	<p>Repos</p>	<p>Appelle un processus de repos : place le robot dans une position accroupie puis détend toutes les articulations.</p>


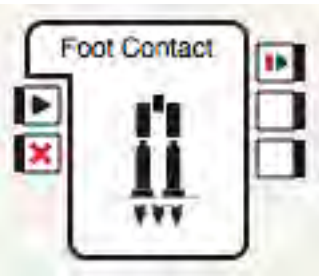
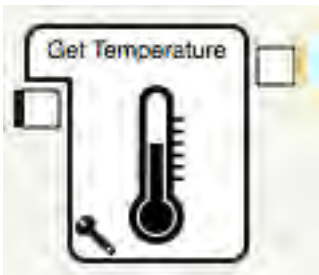

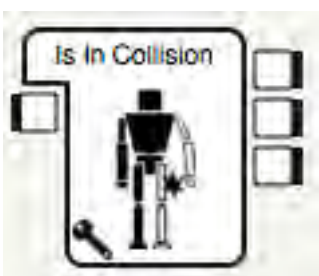
	<h2>Réveil</h2>	<p>Appelle un processus de réveil : raidit toutes les articulations et place le robot dans une position debout (<i>stand Init</i>).</p>
	<h2>S'asseoir</h2>	<p>Le robot tente de s'asseoir (nombre défini d'essais) à partir de n'importe quelle position.</p> <p>Vous pouvez définir le nombre 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.</p> <p>Vous pouvez définir le nombre d'essais dans les paramètres.</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>Faire marcher le robot dans une direction définie dans les paramètres.</p> <p>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>

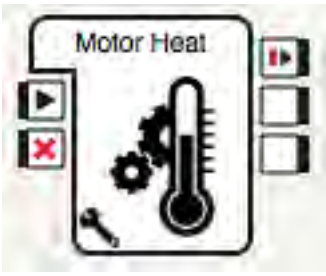
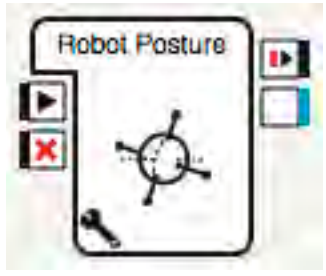
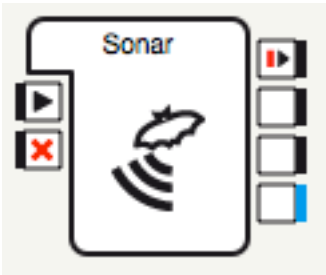

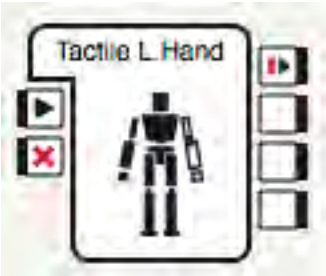
	<h2>Éviter les obstacles</h2>	<p>Marche vers l'avant et tourne à droite lorsqu'il y a un obstacle.</p>
---	-------------------------------	--

SENSING (DÉTECTION)



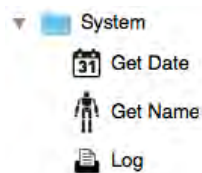
	<h2>Piles</h2>	<p>Vérifie le détecteur des piles. Va stimuler l'<i>output onLow</i> lorsque le niveau des piles est très bas.</p>
	<h2>Amortisseurs</h2>	<p>Vérifie les détecteurs des amortisseurs. Stimule l'<i>output onLeft</i> (gauche) ou <i>onRight</i> (droite) en fonction de l'amortisseur qui est stimulé.</p>


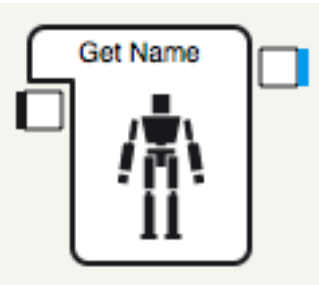
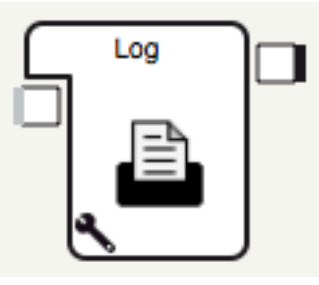
	<p>Détecteur de chute</p>	<p>Détecte si le robot est tombé et s'il a activé le processus de gestion de chute.</p>
	<p>Contact des pieds</p>	<p>Détecte des changements dans le contact des pieds avec le sol.</p>
	<p>Obtenir la température</p>	<p>Détecte la température du matériel (équipement du robot) défini dans les paramètres.</p>
	<p>As-tu ce matériel ?</p>	<p>Si le robot détient le matériel (équipement) défini dans les paramètres, l'<i>output true</i> (vrai) est stimulé. Sinon, c'est l'<i>output false</i> (faux) qui est stimulé.</p>
	<p>Est-ce qu'il y a une collision ?</p>	<p>Vérifie si la partie du corps sélectionnée dans les paramètres est en collision avec une autre partie du corps.</p>

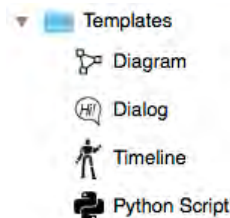
	<p>Température du moteur</p>	<p>Vérifie la température du moteur ou d'un groupe de moteurs et notifie lorsqu'il devient chaud ou froid.</p>
	<p>Posture du robot</p>	<p>Renvoie le nom de la posture du robot chaque fois que la valeur change.</p>
	<p>Sonar</p>	<p>Détecte les obstacles à l'aide de détecteurs ultrasoniques.</p>
	<p>Détecteur tactile de la tête</p>	<p>Détecte un toucher sur les détecteurs tactiles de la tête.</p>
	<p>Détecteur tactile de la main gauche</p>	<p>Détecte un toucher sur les détecteurs tactiles de la main gauche.</p>

	<h2>Détecteur tactile de la main droite</h2>	<p>Détecte un toucher sur les détecteurs tactiles de la main droite.</p>
---	--	--

SYSTEM (SYSTÈME)

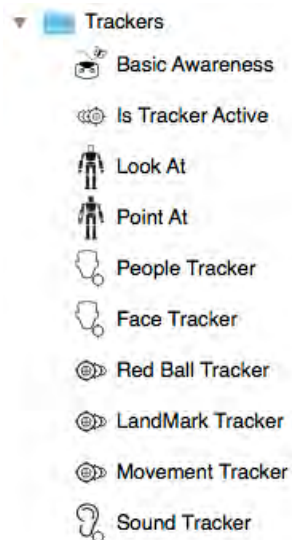


	<h2>Obtenir la date</h2>	<p>Indique la date et l'heure locale.</p>
	<h2>Obtenir le nom</h2>	<p>Renvoie le nom du robot.</p> <p>Note : vous pouvez changer le nom du robot sur sa page Web (adresse IP).</p>
	<h2>Message</h2>	<p>Affiche un message dans le fenêtre de débogage.</p>

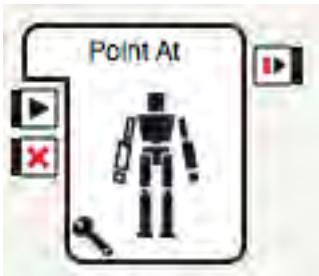



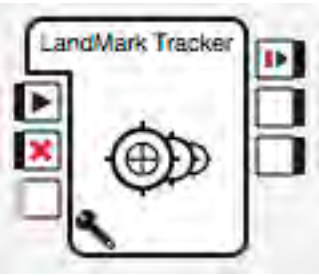


	<h2>Diagramme</h2>	<p>Cette boîte est vide et devrait être utilisée pour créer une boîte diagramme de votre choix.</p> <p>Double-cliquer sur la boîte pour la définir.</p>
	<h2>Dialogue</h2>	<p>Cette boîte est vide et devrait être utilisée pour jouer le sujet de dialogue de votre choix à partir du projet.</p> <p>Double-cliquer sur la boîte pour la définir.</p>
	<h2>Ligne du temps</h2>	<p>Cette boîte est vide et devrait être utilisée pour créer l'animation de votre choix.</p> <p>Double-cliquer sur la boîte pour la définir.</p>
	<h2>Langage scripte <i>Python</i> (codage)</h2>	<p>Cette boîte est vide et devrait être utilisée pour créer une boîte en langage script (<i>Python</i>) de votre choix.</p> <p>Double-cliquer sur la boîte pour la définir.</p>

TRACKERS (DÉTECTEURS)



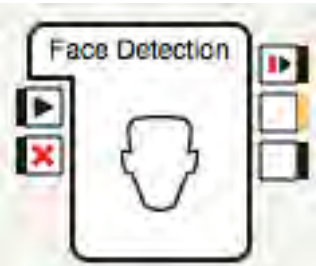
	<h3>Conscience de base</h3>	<p>Cette boîte est une façon simple d'établir et de maintenir un contact visuel avec les gens.</p>
	<h3>Le détecteur est-il actif ?</h3>	<p>Vérifie si le détecteur est actif. L'<i>output Yes</i> est stimulé si le détecteur est actif, et <i>No</i> s'il ne l'est pas.</p>
	<h3>Regarder vers...</h3>	<p>Fait regarder le robot vers l'endroit désiré.</p>

	<p>Pointer vers...</p>	<p>Fait pointer le robot vers l'endroit désiré.</p>
	<p>Détecteur de personnes</p>	<p>Fait détecter et suivre les gens avec différents modes.</p>
	<p>Détecteur de visage</p>	<p>Fait détecter et suivre les visages avec différents modes.</p>
	<p>Détecteur de balle rouge</p>	<p>Fait détecter et suivre une balle rouge avec différents modes.</p>
	<p>Détecteur de point de repère</p>	<p>Fait détecter et suivre un point de repère avec différents modes.</p>


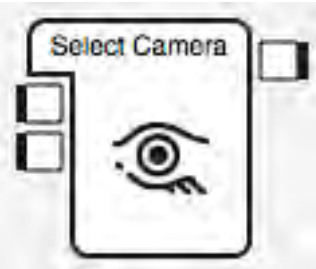


	<h2>Détecteur de mouvement</h2>	<p>Fait tourner la tête du robot dans la direction où il y a le plus de mouvements.</p>
	<h2>Détecteur de son</h2>	<p>Fait détecter et suivre un son avec différents modes.</p>

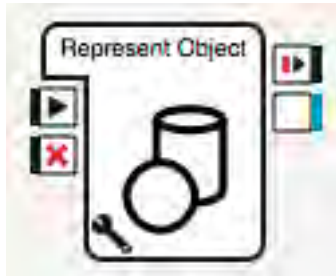
VISION

- ▼ Vision
 - 👤 Face Detection
 - 🔍 Face Reco.
 - 💡 Is Backlit
 - 👤 Is In Darkness
 - 👤 Learn Face
 - 🔍 NAOMark
 - 📹 Record Video
 - 👁️ Select Camera
 - 📷 Take Picture
 - 🗑️ Unlearn All Faces
 - 🗑️ Unlearn Face
 - 🔍 Vision Reco.

	<h2>Détection faciale</h2>	<p>Détecte le visage de gens et renvoie le nombre de visages détectés.</p>
---	----------------------------	--

	<p>Reconnaissance faciale</p>	<p>Détecte le visage des gens et reconnaît ceux qui sont connus du robot (qui ont été appris, voir <i>Apprendre un visage</i>).</p>
	<p>L'image est-elle en contre-jour ?</p>	<p>Vérifie si l'image des caméras est en contre-jour.</p>
	<p>Suis-je dans l'obscurité ?</p>	<p>Vérifie, en utilisant les caméras, si le robot est dans l'obscurité.</p>
	<p>Apprendre un visage</p>	<p>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.</p>
	<p>«NAO marques»</p>	<p>Détecte les <i>NAO marques</i> et renvoie le numéro d'identification de celles-ci.</p>

	<p>Enregistrer une vidéo</p>	<p>Enregistre une vidéo à partir de l'une des caméras. Arrête l'enregistrement lorsque la durée définie dans les paramètres est terminée.</p>
	<p>Sélectionner une caméra</p>	<p>Sélectionne la caméra à utiliser.</p>
	<p>Prendre une photo</p>	<p>Prend une photographie à l'aide d'une des caméras.</p>
	<p>Oublier tous les visages</p>	<p>Supprime tous les visages appris et enregistrés dans la banque de données.</p>
	<p>Oublier des visages</p>	<p>Supprimer les visages associés au(x) nom(s) défini(s) dans l'input.</p>
	<p>Reconnaissance visuelle</p>	<p>Reconnaît des images, des objets ou des lieux qui sont connus du robot (qui ont été appris).</p>



**Ajouter ou enlever
un objet**

Permet d'ajouter ou d'enlever
un objet de la *représentation du
monde* du robot.