

Nao est un **robot humanoïde** de 58 cm de haut, programmable développé par la société française Aldebaran Robotics. Cette société française a été créée en 2005.



Il est équipé de **26 articulations motorisées**, qui lui permettent de se déplacer et de saisir des objets.

Nao est équipé de **deux caméras**, **deux microphones**, qui lui permettent de voir et d'entendre « en trois dimensions », ainsi que de **haut-parleurs pour s'exprimer**, de capteurs ultrasons, de **senseurs tactiles**, de capteurs infrarouges, etc. Il peut se connecter en WiFi.

Caractéristiques techniques de NAO :

Hauteur : 58 cm

Poids : 4,8 kg

Autonomie : 90 min

Degrés de liberté : 26

Processeur Intel ATOM 1,6GHz

Système d'exploitation intègre Linux

Systèmes d'exploitations : compatibles Windows, Mac OS,
Linux

Langages de programmation : C++, Python, JAVA, MATLAB,
Urbi, C, Connectivite : internet, Wi-Fi



Nao ne sait rien faire au moment de sa livraison.

Son potentiel d'apprentissage est sans limite.

Pour les développeurs: ils peuvent stocker dans la mémoire de Nao des applications qui lui permettent d'exécuter un nombre croissant de tâches en autonomie : tenir une conversation, porter des objets, faire des recherches sur Internet, tout est possible avec Nao.



Programmation d'un NAO : Chorégraphe

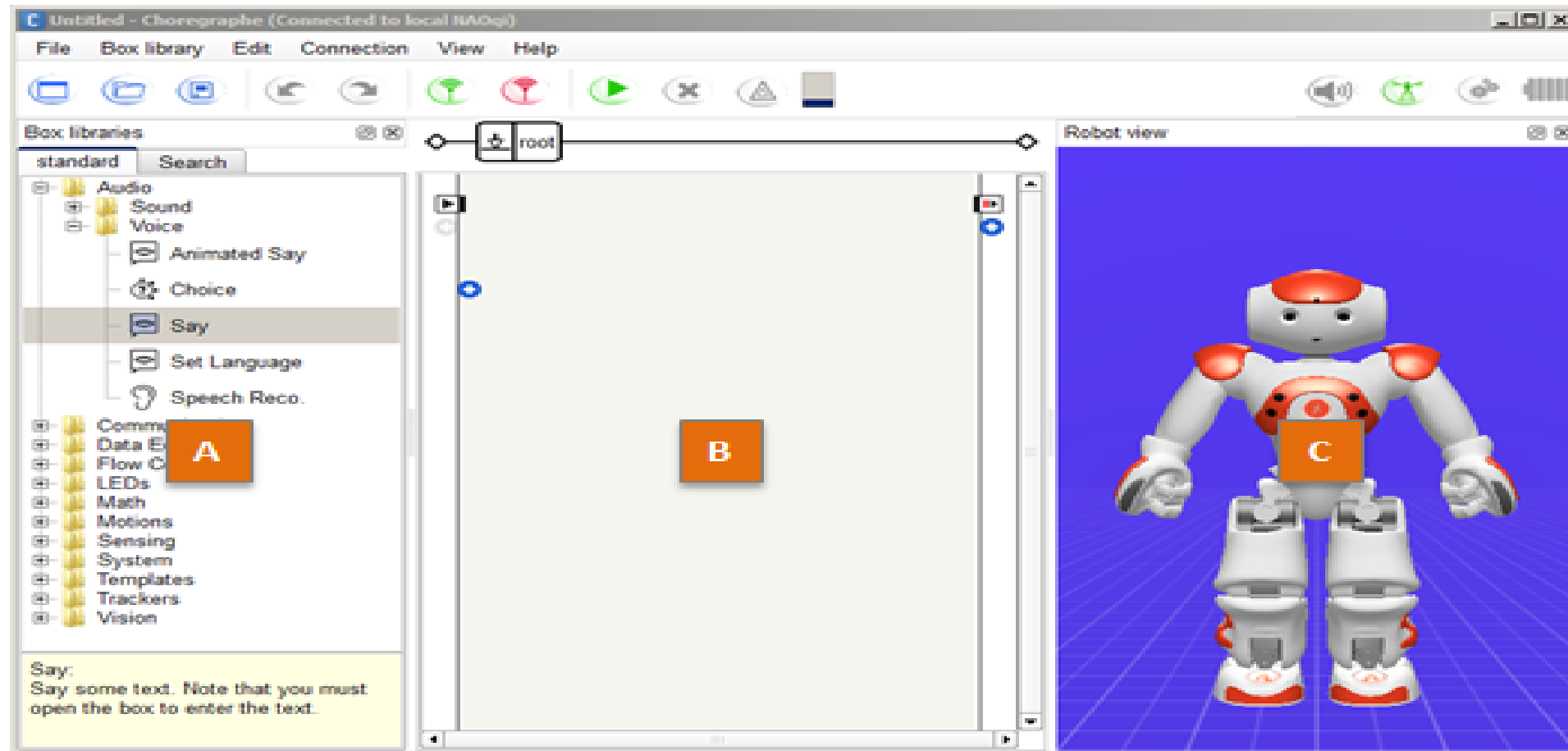
- « **Chorégraphe** » est le logiciel de programmation fourni par le constructeur de **NAO**.
- Pour programmer un NAO, il suffit de relier, en **série** ou en **parallèle**, différentes boites de commande, chaque boite renferme une **action à exécuter** par le NAO.
- On distingue deux types de boite de commande:
 - Boites de commande renfermant un langage **python**
 - Boites de commande de types « Animation » permettant **d'enregistrer la position dans le temps des différents moteurs** (et donc permettre au NAO de reproduire un mouvement).

Contrôle des moteurs

« **Chorégraphie** » permet d'enregistrer la position des 26 moteurs en fonction du temps. Pour un mouvement enregistré, nous disposons de 26 courbes, chaque courbe traduit le déplacement et la rotation dans le temps d'un moteur. Il est possible de retravailler manuellement chaque courbe (points par points) afin d'optimiser le mouvement dans l'espace et de le rendre le plus naturel possible.

Moteurs	Degrés de liberté	Nombre
Tête	2	1
Épaule	2	2
Coude	2	2
Main	2	2
Hanche	2	2
Genou	1	2
Cheville	2	2

Fenêtre principale du logiciel chorégraphe



A : c'est la fenêtre pour accéder aux librairies des boîtes de commandes.

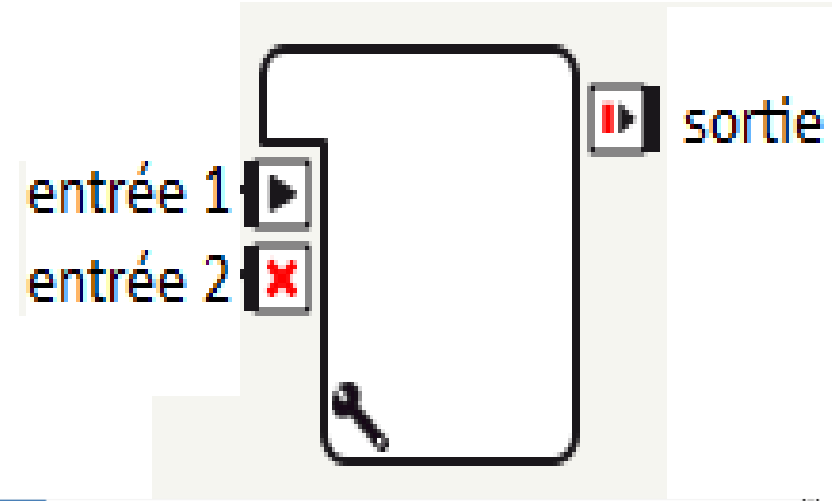
Ces boîtes sont triées par leur catégorie d'action (mouvement, voyants, etc.)

B : c'est la fenêtre où sont ajoutées les actions que NAO va effectuer.

Les actions sont liées entre elles par des liens, une action à une entrée et une sortie.

C : C'est la vue de NAO en 3D, elle simule le robot virtuellement pour avoir un pré-rendu du programme.

Une **boîte de commande** est composée de deux entrées et d'une sortie, lorsqu'une de ses entrées est activée par un signal extérieur, le programme associé a cette entrée est exécuté. On retrouve en effet dans le code python un espace de programmation dédié a chaque entrée.



Def onInput_onStart(self): les lignes de codes associés sont exécutées si un signal d'entrée est détecté sur l'entrée 1.

Def onInput_onStop(self): les lignes de codes sont exécutées si un signal d'entrée est détecté sur l'entrée 2.

L'activation de la sortie est générée une fois le programme termine.

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

    def onLoad(self):
        #~ puts code for box initialization here
        pass

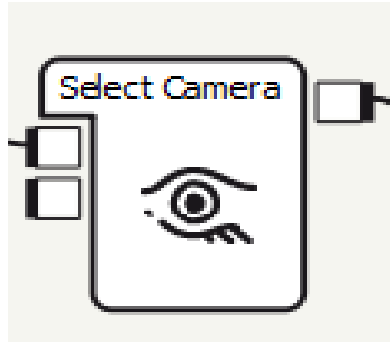
    def onUnload(self):
        #~ puts code for box cleanup here
        pass

    def onInput_onStart(self):
        #~ self.onStopped() #~ activate output of the box
        pass

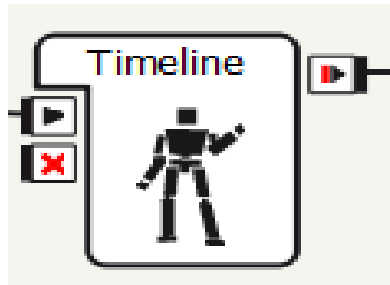
    def onInput_onStop(self):
        self.onUnload() #~ it is recommended to call onUnload of
        this box in a onStop method, as the code written in onUnload is
        used to stop the box as well
        pass
```



Permet au Nao de se mettre en position debout.



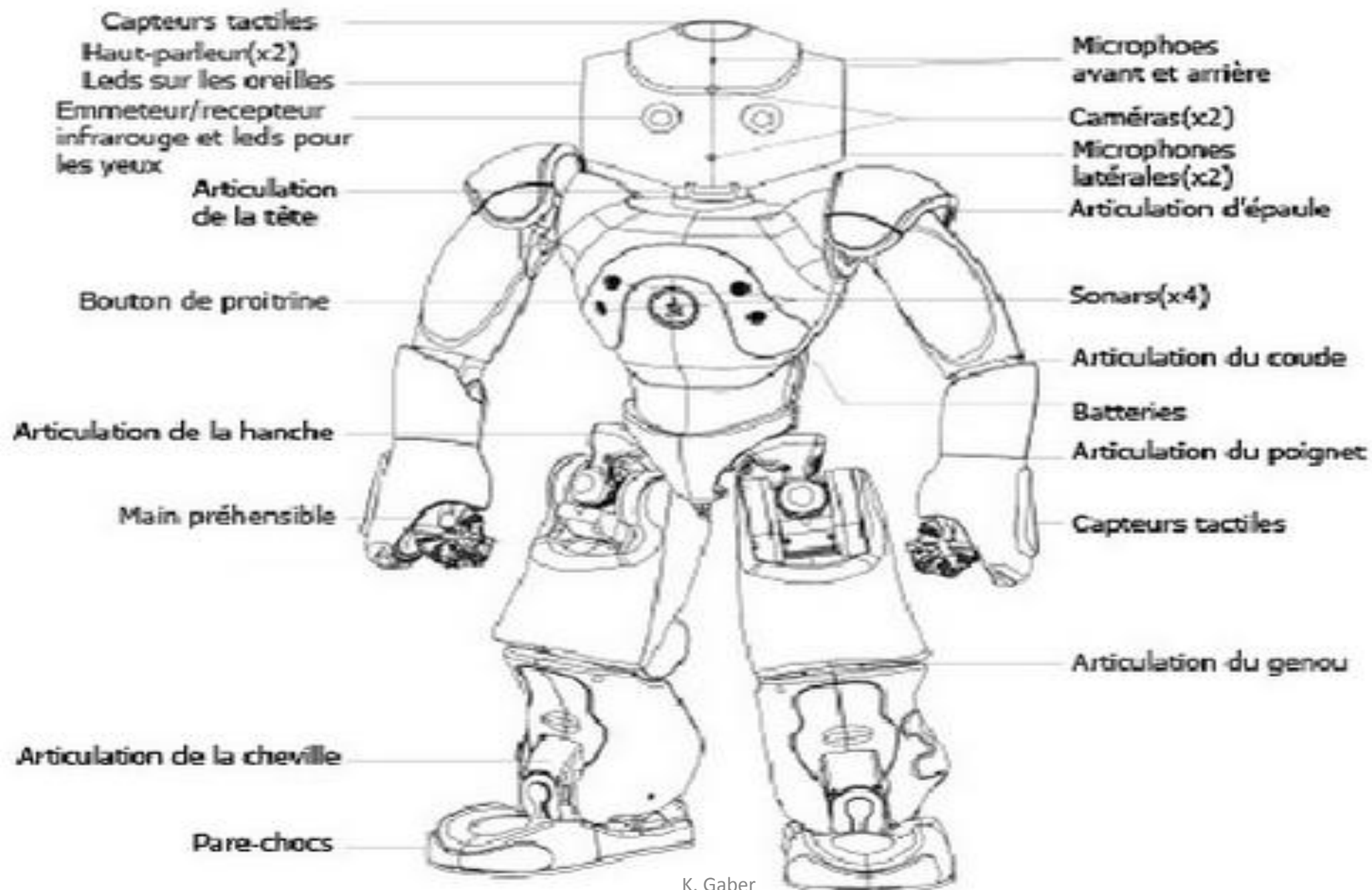
permet de sélectionner la caméra par laquelle Nao va pouvoir voir.



Permet d'enregistrer un enchainement de mouvement en fonction du temps.



permet de reconnaître un élément rouge et de se déplacer vers celle-ci automatiquement.



TP NAO - Détecter un objet

Ecrire un programme qui permette à Nao de repérer un objet, se diriger vers lui, et le ramasser.