



NIRYO

HUMAN - MOTION - ROBOT

VISION

1

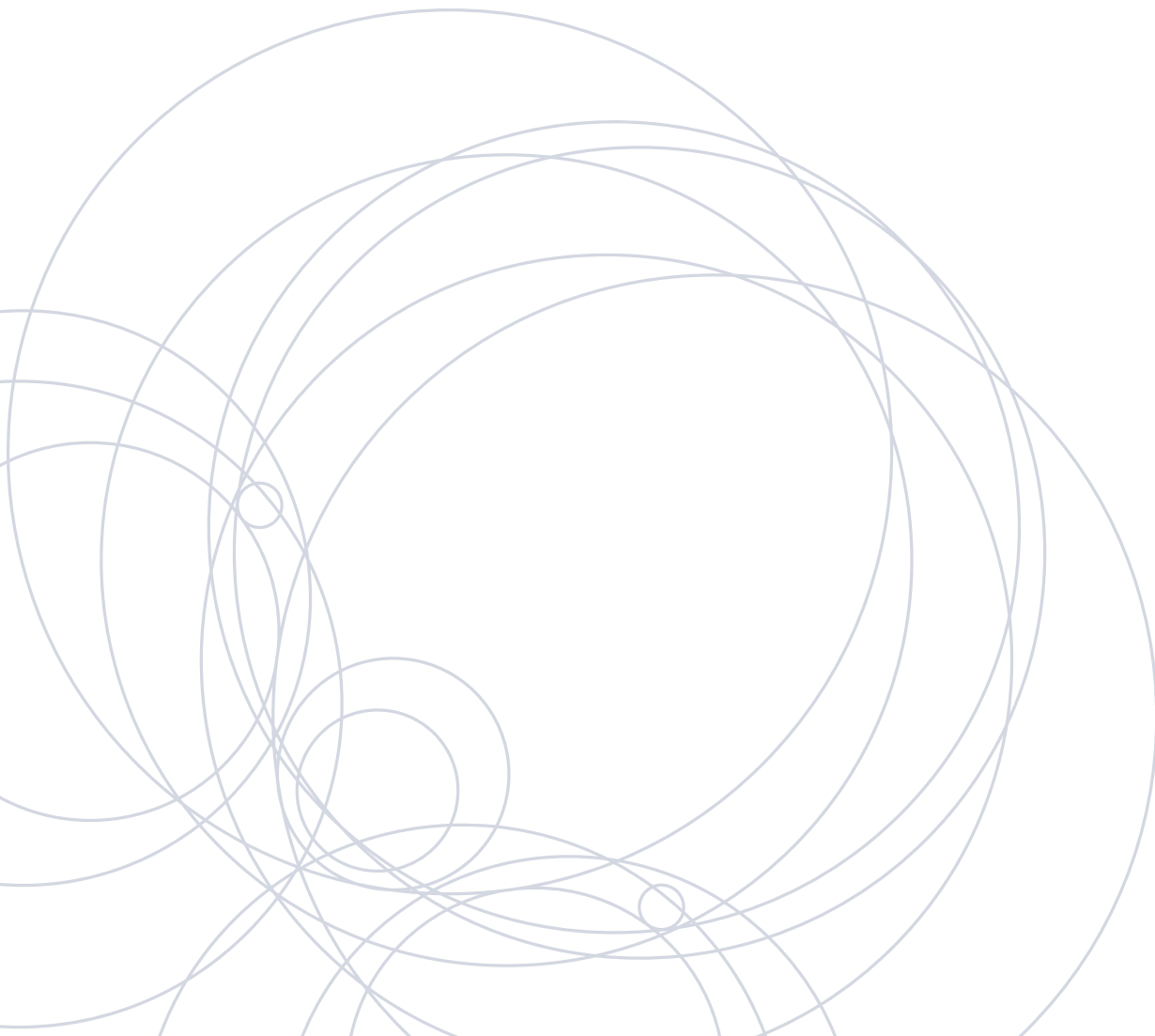


Découverte de la vision avec



Blockly

OBJECTIFS	3
PRÉREQUIS	3
CE DONT VOUS AUREZ BESOIN	3
INTRODUCTION	3
APPLICATIONS	4
EXERCICE 1 : VISION PICK	4
EXERCICE 2 : CONDITIONNEMENT MULTI-RÉFÉRENCES	6
POUR ALLER PLUS LOIN	7
CORRECTION	8



OBJECTIFS

- Découvrir les fonctions de vision du Niryo One disponibles dans Blockly
- Apprendre à réaliser un procédé industriel en utilisant la vision

PRÉREQUIS

- Connaissances de base en Blockly
- Avoir pris connaissance du manuel utilisateur du Set Vision

CE DONT VOUS AUREZ BESOIN

- Un ordinateur équipé du WIFI ou d'un port Ethernet
- Niryo One Studio version ≥ 2.3
- Software Niryo One version ≥ 2.3
- Un Niryo One avec un workspace fixé
- Les 6 objets manipulables fournis avec le Set Vision

INTRODUCTION

Le Set Vision est un système de Vision intégré au Niryo One permettant au robot de fonctionner de manière autonome.

En effet, tous les calculs algorithmiques sont effectués sur le robot ce qui permet notamment d'enregistrer des séquences et de les reproduire sans avoir à connecter le Niryo One à un ordinateur.

Les instructions permettant l'utilisation du Set Vision avec Niryo One Studio sont détaillées dans le manuel d'utilisation du produit.



AVANT DE POURSUIVRE

Avant de passer aux applications, s'assurer que :

- Le robot est calibré,
- Le workspace qui va être utilisé est bien fixé et défini dans Niryo One Studio,
- La qualité de connexion est suffisante pour le stream vidéo (une connexion Ethernet fournira une qualité optimale).

EXERCICE 1 : VISION PICK

1. Placement en position d'observation

Pour effectuer le traitement d'images, la caméra du Niryo One a besoin de se trouver dans une position fournissant une vue dégagée du workspace.

Utiliser les blocs de mouvement pour amener la caméra à la position idéale pour observer le workspace.

2. Effectuer son premier Vision Pick

Le bloc Vision Pick permet de réaliser un visual picking en une seule action. Il retourne un booléen indiquant si, oui ou non, un objet a été détecté et manipulé.

Après avoir vérifié que l'outil du Niryo One est bien sélectionné dans Niryo One Studio, poser un objet du Set Vision sur le workspace.

Utiliser le bloc Vision Pick pour effectuer le visual picking de l'objet, et sauvegarder le booléen au sein d'une variable.

3. Effectuer une action en fonction du résultat du bloc

Maintenant que la base du programme est définie, nous allons ajouter des actions à effectuer autour de cette phase de picking. Nous souhaitons, après avoir saisi l'objet, effectuer un mouvement jusqu'à une position pour l'y relâcher.

Incrémenter le programme de sorte que, si le Vision Pick a fonctionné, le robot aille déposer l'objet à une position donnée, sinon, activer le mode apprentissage.

Astuce :

Pour cette action de place, il est possible d'utiliser le bloc Place from pose.

4. Attendre qu'un objet soit saisissable

Dans certains cas, il est possible que la séquence soit lancée alors que le workspace ne présente aucun objet à manipuler.

Créer une boucle qui attend qu'un objet soit détecté et manipulé pour effectuer le relâcher à une autre position.

5. Effectuer des picks tant que cela est possible

Dans la majorité de nombreux cas de picking, un robot a plus d'un objet à attraper. Il est donc possible d'imaginer une boucle dans laquelle le robot effectuera des Vision Picks tant que cela est possible.

Créer une boucle qui se charge de faire un pick and place tant que cela est possible puis active le mode apprentissage une fois qu'il n'y a plus d'objets à manipuler.

Astuces :

La position à laquelle le robot doit relâcher les objets peut rester identique au fur et à mesure des itérations. Ceci dit, veiller à ce que le robot n'entre pas en collision avec les objets précédemment posés.

Préférer une position éloignée du workspace et des objets pour activer le mode apprentissage afin d'éviter tout contact avec la surface de travail et les objets.

EXERCICE 2 : CONDITIONNEMENT MULTI-RÉFÉRENCES

Maintenant que le fonctionnement du bloc Vision Pick est compris, nous allons chercher à ranger les objets sur le côté de l'espace de travail en fonction de leur forme.

1. Effectuer une action en fonction de la forme de l'objet

Dans cette situation, votre workspace contient deux objets de formes différentes : un carré et un rond.

Il est attendu que le robot dépose l'objet de la première forme à gauche du workspace, et l'objet de la seconde de l'autre côté.

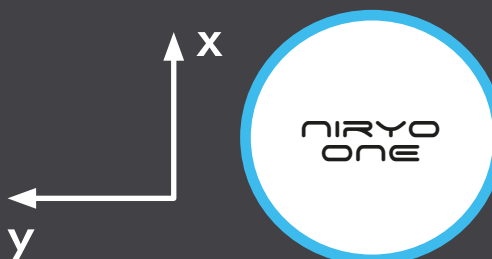
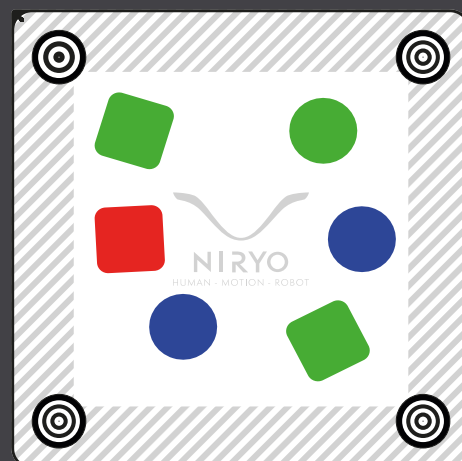
Réaliser cette tâche en utilisant le bloc Place From Pose pendant les phase de place.

2. Effectuer un conditionnement des objets en fonction de leur forme

Maintenant que nous avons mis en place un système de condition pour placer les objets à différentes positions selon leur forme, nous allons simuler un conditionnement.

Pour ce faire, nous allons utiliser les six objets du Set vision, à savoir trois carrés et trois ronds. Il est attendu que le robot les positionne d'un côté ou de l'autre du workspace, comme précédemment, en les alignant de chaque avec une même valeur de y.

Créer un script qui prend les objets carrés pour les ranger à gauche du workspace en les décalant de quelques centimètres selon x, tout en conservant la même valeur de y (voir illustration ci-contre). Faire de même pour les objets ronds, sur la droite du workspace.



POUR ALLER PLUS LOIN

En sachant effectuer un conditionnement via le module de vision, il est possible de réaliser des processus plus complexes.

Par exemple, le Set Vision peut être utilisé dans un environnement de travail présentant un Convoyeur. Dans ce cas, le bloc **Is Object Detected** est particulièrement utile puisqu'il permet de savoir si un objet est entré dans le workspace situé sur le Convoyeur. Il permet alors, si un objet est détecté, de stopper le Convoyeur pour effectuer un Vision Pick.

Pour aller plus loin encore, le Set Vision est polyvalent et peut être contrôlé via l'API TCP Python, qui permet d'effectuer ses propres traitements d'images sur le flux vidéo récupéré par le robot.

Retrouvez tous nos supports pédagogiques sur

www.niryo.com

CORRECTION EXERCICE 1 : VISION PICK

1. Placement en position d'observation

Pour effectuer le traitement d'images, la caméra du Niryo One a besoin de se trouver dans une position fournissant une vue dégagée du workspace.

Utiliser les blocs de mouvement pour amener la caméra à la position idéale pour observer le workspace.

Par exemple, si le workspace est en face du robot, il est possible d'utiliser une commande de mouvement comme Move pose.



Il est possible d'afficher le retour vidéo sur Niryo One Studio via le bouton "Camera streaming" pour s'assurer que les quatre marqueurs sont visibles par la caméra.

2. Effectuer son premier Vision Pick

Le bloc Vision Pick permet de réaliser un visual picking en une seule action. Il retourne un booléen indiquant si, oui ou non, un objet a été détecté et manipulé.

Après avoir vérifié que l'outil du Niryo One est bien sélectionné dans Niryo One Studio, poser un objet du Set Vision sur le workspace.

Utiliser le bloc Vision Pick pour effectuer le visual picking de l'objet, et sauvegarder le booléen au sein d'une variable.

Il est possible de créer une variable, nommée ici "result", pour stocker le résultat du picking.

Dans cet exemple de correction, le workspace utilisé est enregistré sous le nom de "Vision_curriculum", et la position d'observation correspond à un espace de travail spécifique. Ces paramètres sont à ajuster pour chaque situation.


```

Change tool to Standard gripper
Move pose Pose x 0.17 y 0 z 0.3 roll 0 pitch 1.57 yaw 0
set result to Vision pick
  Color ANY
  Shape OBJECT
  in workspace "Vision_curriculum"
  with height offset (mm) 0

```

3. Effectuer une action en fonction du résultat du bloc

Maintenant que la base du programme est définie, nous allons ajouter des actions à effectuer autour de cette phase de picking. Nous souhaitons, après avoir saisi l'objet, effectuer un mouvement jusqu'à une position pour l'y relâcher.

Incrémenter le programme de sorte que, si le Vision Pick a fonctionné, le robot aille déposer l'objet à une position donnée, sinon, activer le mode apprentissage.

Astuce :

Pour cette action de place, il est possible d'utiliser le bloc Place from pose.

```

Change tool to Standard gripper
Move pose Pose x 0.17 y 0 z 0.3 roll 0 pitch 1.57 yaw 0
set result to Vision pick
  Color ANY
  Shape OBJECT
  in workspace "Vision_curriculum"
  with height offset (mm) 0
if result
do
  Place from pose Pose x 0 y 0.15 z 0.15 roll -1.57 pitch 1.57 yaw 0
else
  Activate learning mode

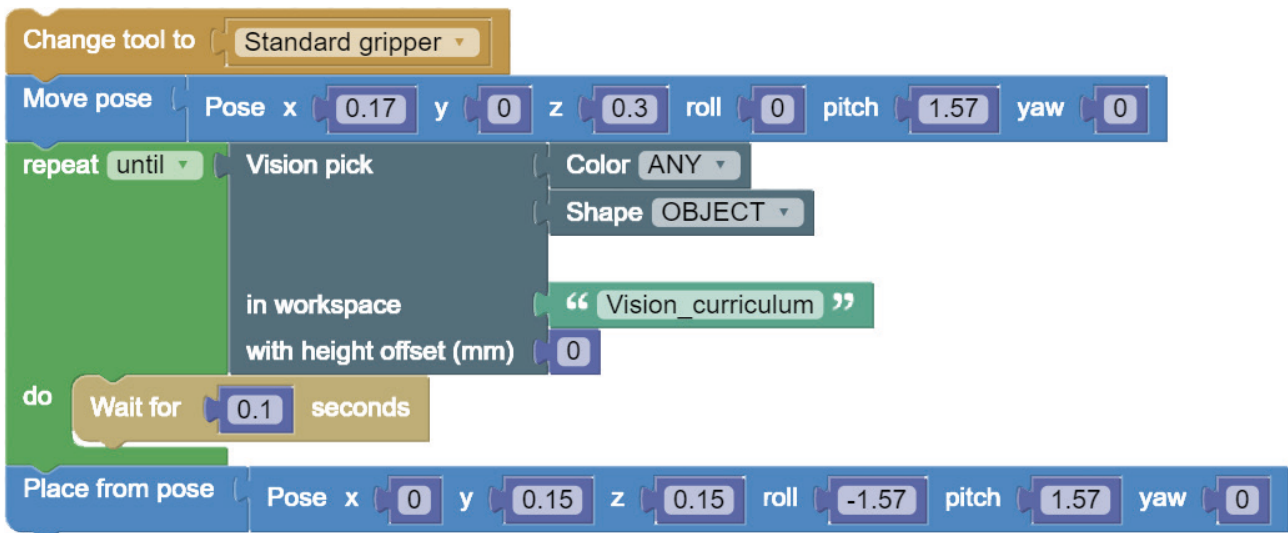
```

Ainsi, si un objet est attrapé pendant la phase de picking, il sera relâché à une autre position. C'est un premier pick & place !

4. Attendre qu'un objet soit saisissable

Dans certains cas, il est possible que la séquence soit lancée alors que le workspace ne présente aucun objet à manipuler.

Créer une boucle qui attend qu'un objet soit détecté et manipulé pour effectuer le relâcher à une autre position.



Une boucle repeat until est ici utilisée afin d'attendre qu'un Vision Pick ait lieu.

Dans cette illustration, la séquence consiste à enchaîner des attentes de 0,1 seconde jusqu'à ce qu'un Vision Pick ait lieu. Après celui-ci, l'objet est, comme précédemment, déposé sur le côté du workspace.

5. Effectuer des picks tant que cela est possible

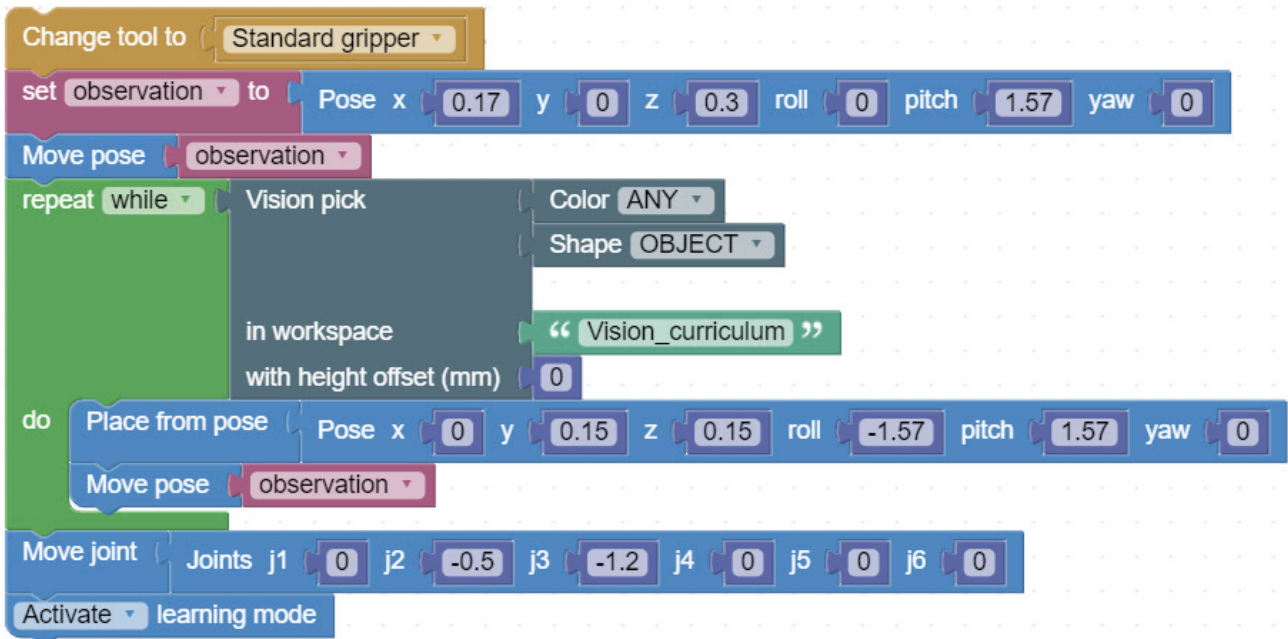
Dans la majorité de nombreux cas de picking, un robot a plus d'un objet à attraper. Il est donc possible d'imaginer une boucle dans laquelle le robot effectuera des Vision Picks tant que cela est possible.

Créer une boucle qui se charge de faire un pick and place tant que cela est possible puis active le mode apprentissage une fois qu'il n'y a plus d'objets à manipuler.

Astuces :

La position à laquelle le robot doit relâcher les objets peut rester identique au fur et à mesure des itérations. Ceci dit, veiller à ce que le robot n'entre pas en collision avec les objets précédemment posés.

Préférer une position éloignée du workspace et des objets pour activer le mode apprentissage afin d'éviter tout contact avec la surface de travail et les objets.



```
Change tool to Standard gripper
set observation to Pose x 0.17 y 0 z 0.3 roll 0 pitch 1.57 yaw 0
Move pose observation
repeat while Vision pick
  Color ANY
  Shape OBJECT
  in workspace "Vision_curriculum"
  with height offset (mm) 0
do
  Place from pose Pose x 0 y 0.15 z 0.15 roll -1.57 pitch 1.57 yaw 0
  Move pose observation
Move joint Joints j1 0 j2 -0.5 j3 -1.2 j4 0 j5 0 j6 0
Activate learning mode
```

Avec ce programme, chaque Vision Pick & Place possible est réalisé. Une fois que le workspace ne contient plus d'objets à saisir, le robot effectue un move joint et passe en mode apprentissage.

À noter : dans cet exemple de correction, les objets sont tous placés au même endroit.

CORRECTION EXERCICE 2 : CONDITIONNEMENT MULTI-RÉFÉRENCES

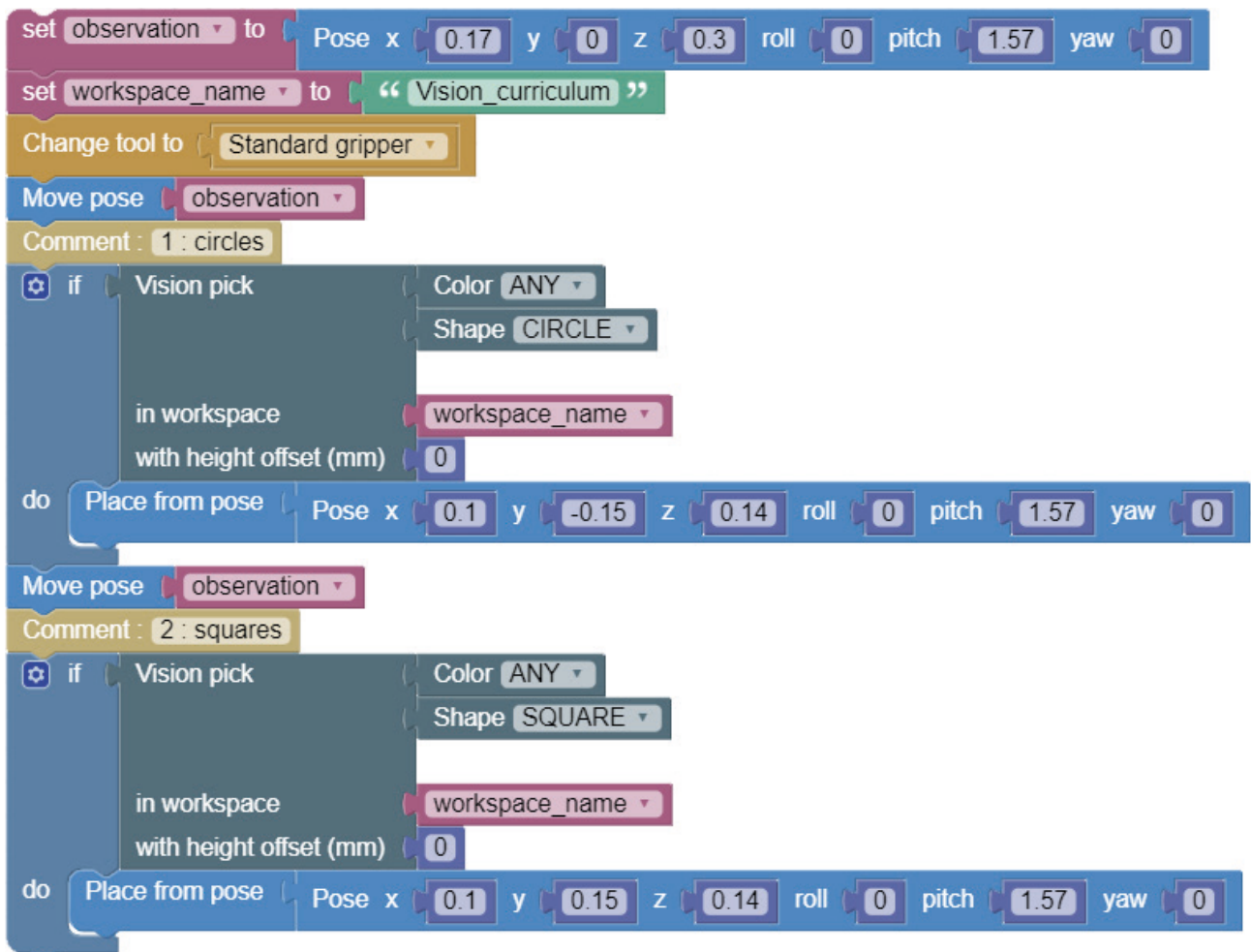
Maintenant que le fonctionnement du bloc Vision Pick est compris, nous allons chercher à ranger les objets sur le côté de l'espace de travail en fonction de leur forme.

1. Effectuer une action en fonction de la forme de l'objet

Dans cette situation, votre workspace contient deux objets de formes différentes : un carré et un rond.

Il est attendu que le robot dépose l'objet de la première forme à gauche du workspace, et l'objet de la seconde de l'autre côté.

Réaliser cette tâche en utilisant le bloc Place From Pose pendant les phase de place.



```
set observation to Pose x 0.17 y 0 z 0.3 roll 0 pitch 1.57 yaw 0
set workspace_name to "Vision_curriculum"
Change tool to Standard gripper
Move pose observation
Comment : 1 : circles
if Vision pick
  Color ANY
  Shape CIRCLE
  in workspace workspace_name
  with height offset (mm) 0
do Place from pose Pose x 0.1 y -0.15 z 0.14 roll 0 pitch 1.57 yaw 0
Move pose observation
Comment : 2 : squares
if Vision pick
  Color ANY
  Shape SQUARE
  in workspace workspace_name
  with height offset (mm) 0
do Place from pose Pose x 0.1 y 0.15 z 0.14 roll 0 pitch 1.57 yaw 0
```

Premièrement, plusieurs variables sont définies comme le nom du workspace qui sera utilisé ainsi que la position d'observation.

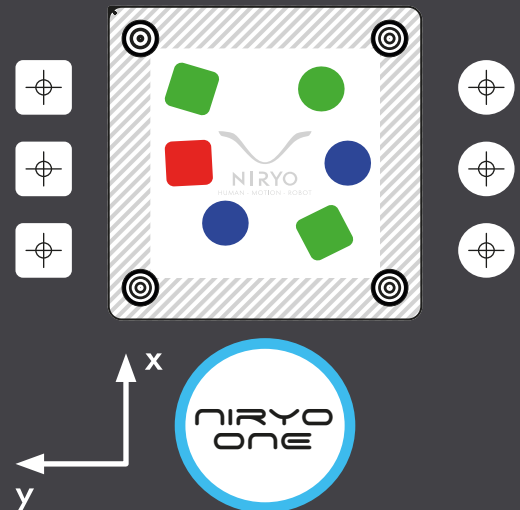
Ensuite, le robot se déplace jusqu'à cette position et va déplacer l'ensemble des objets présents sur le workspace, en commençant par les carrés, qui sont ensuite placés à gauche du workspace, puis les ronds, qui sont ensuite placés à sa droite.

2. Effectuer un conditionnement des objets en fonction de leur forme

Maintenant que nous avons mis en place un système de condition pour placer les objets à différentes positions selon leur forme, nous allons simuler un conditionnement.

Pour ce faire, nous allons utiliser les six objets du Set vision, à savoir trois carrés et trois ronds. Il est attendu que le robot les positionne d'un côté ou de l'autre du workspace, comme précédemment, en les alignant de chaque côté avec une même valeur de y.

Créer un script qui prend les objets carrés pour les ranger à gauche du workspace en les décalant de quelques centimètres selon x, tout en conservant la même valeur de y (voir illustration ci-contre). Faire de même pour les objets ronds, sur la droite du workspace.



La base du programme est ici la même que la précédente.

Ici, on ajoute le calcul de la valeur de x à chaque itération grâce à la variable "new_x_pos", en choisissant un décalage de 5 centimètres à chaque nouvelle itération, en basant le calcul sur la variable "count_catch", qui est utilisée pour stocker le nombre d'objets collectés (voir l'illustration page suivante).

```

set observation to Pose x 0.17 y 0 z 0.3 roll 0 pitch 1.57 yaw 0
set workspace_name to "Vision_curriculum"
Change tool to Standard gripper
Move pose observation
Comment : 1 : circles
set count_catch to 0
repeat while Vision pick
  Color ANY
  Shape CIRCLE
  in workspace workspace_name
  with height offset (mm) 0
do
  set new_x_pos to 0.1 + 0.05 * count_catch
  Place from pose Pose x new_x_pos y -0.15 z 0.14 roll 0 pitch 1.57 yaw 0
  set count_catch to count_catch + 1
Move pose observation
Comment : 2 : squares
set count_catch to 0
repeat while Vision pick
  Color ANY
  Shape SQUARE
  in workspace workspace_name
  with height offset (mm) 0
do
  set new_x_pos to 0.1 + 0.05 * count_catch
  Place from pose Pose x new_x_pos y 0.15 z 0.14 roll 0 pitch 1.57 yaw 0
  set count_catch to count_catch + 1

```