Workshop 4:
From Simulation to real world

Follow this step-by-step guide during the session.

Step 0 – Goal of the Workshop

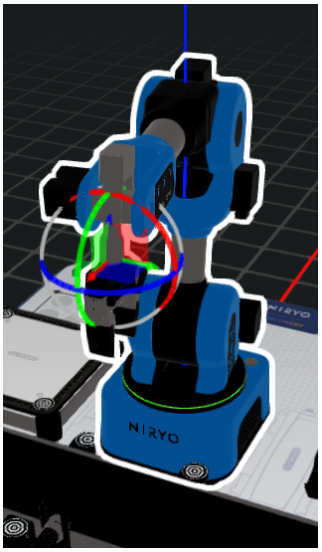
You will learn how to:

- Build a complete pick-and-place program **inside the simulation**
- Transfer this program to the **real robot**
- Correct and adapt poses so the robot behaves safely in the physical world
- Use key features: conveyor, IR sensor, vision workspaces, and Vision Pick

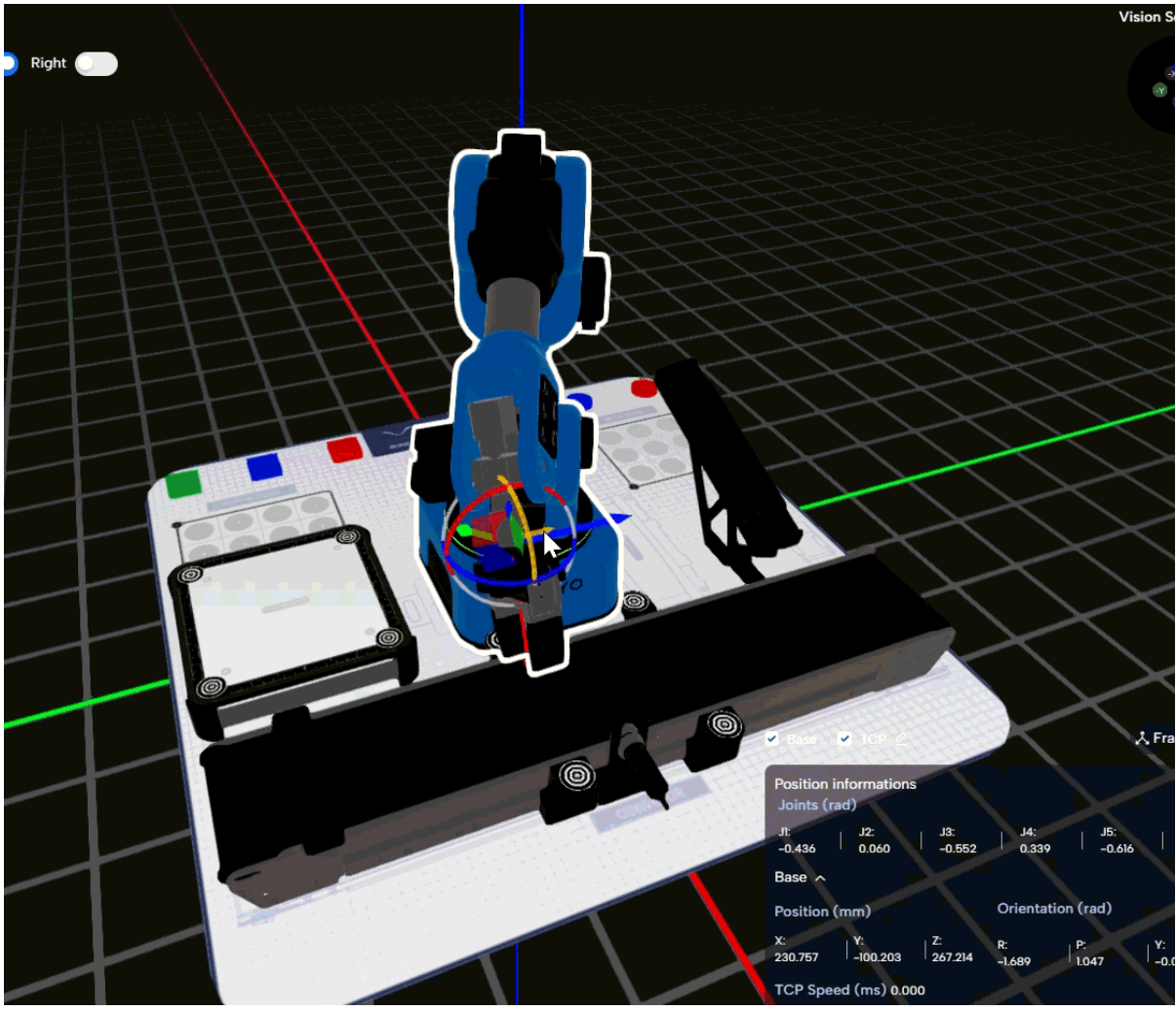
---

Step 1: Getting Started with the Simulation

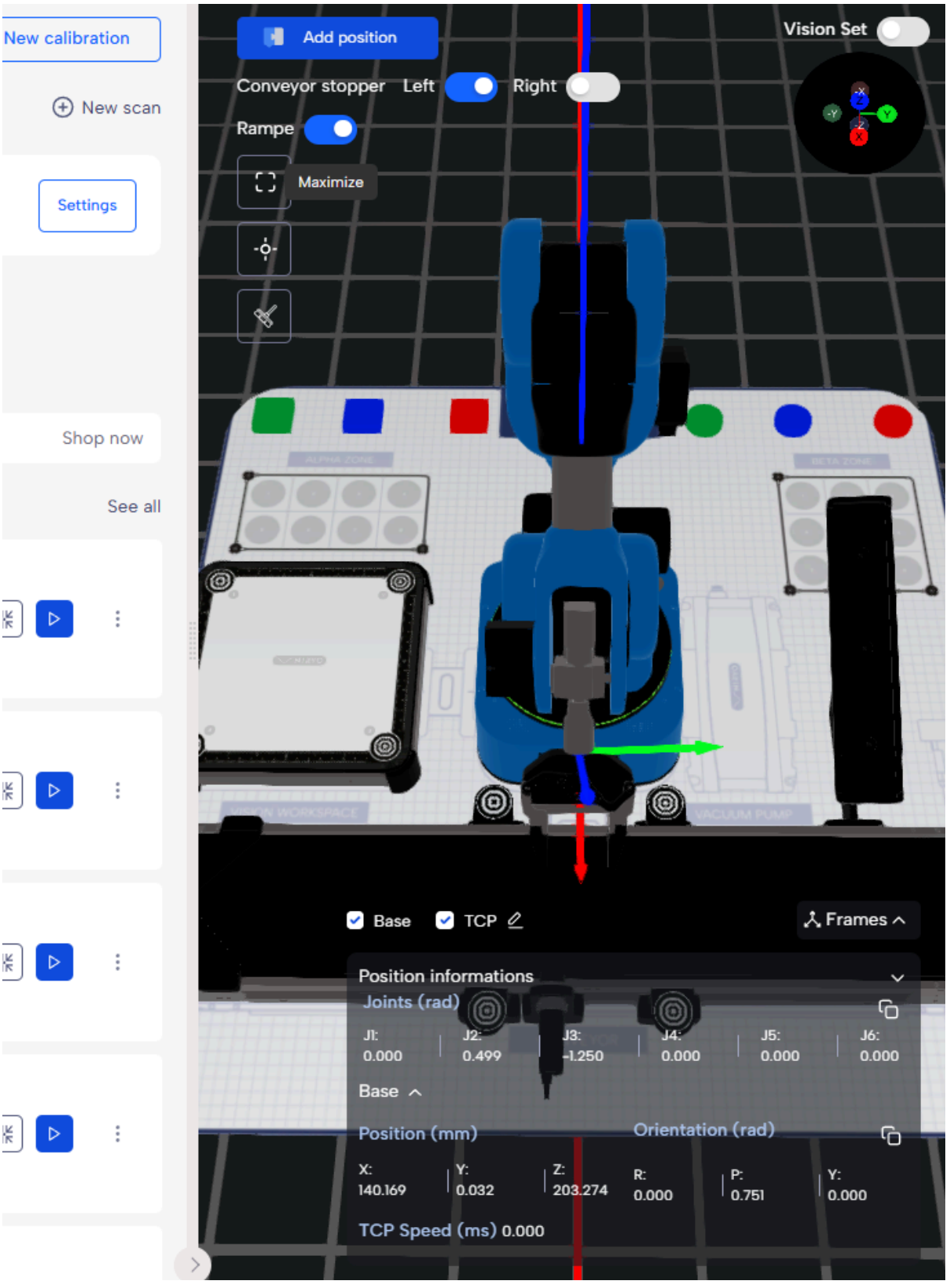
- Click "**Launch Simulation**" from NiryoStutio
- Click on the robot in the simulation — a **gizmo** will appear on the end effector.



- Use this gizmo to move the robot:
    - Drag **a single axis arrow** to move linearly along that axis.
    - Drag **a plane** to move the robot along the corresponding XY plane.



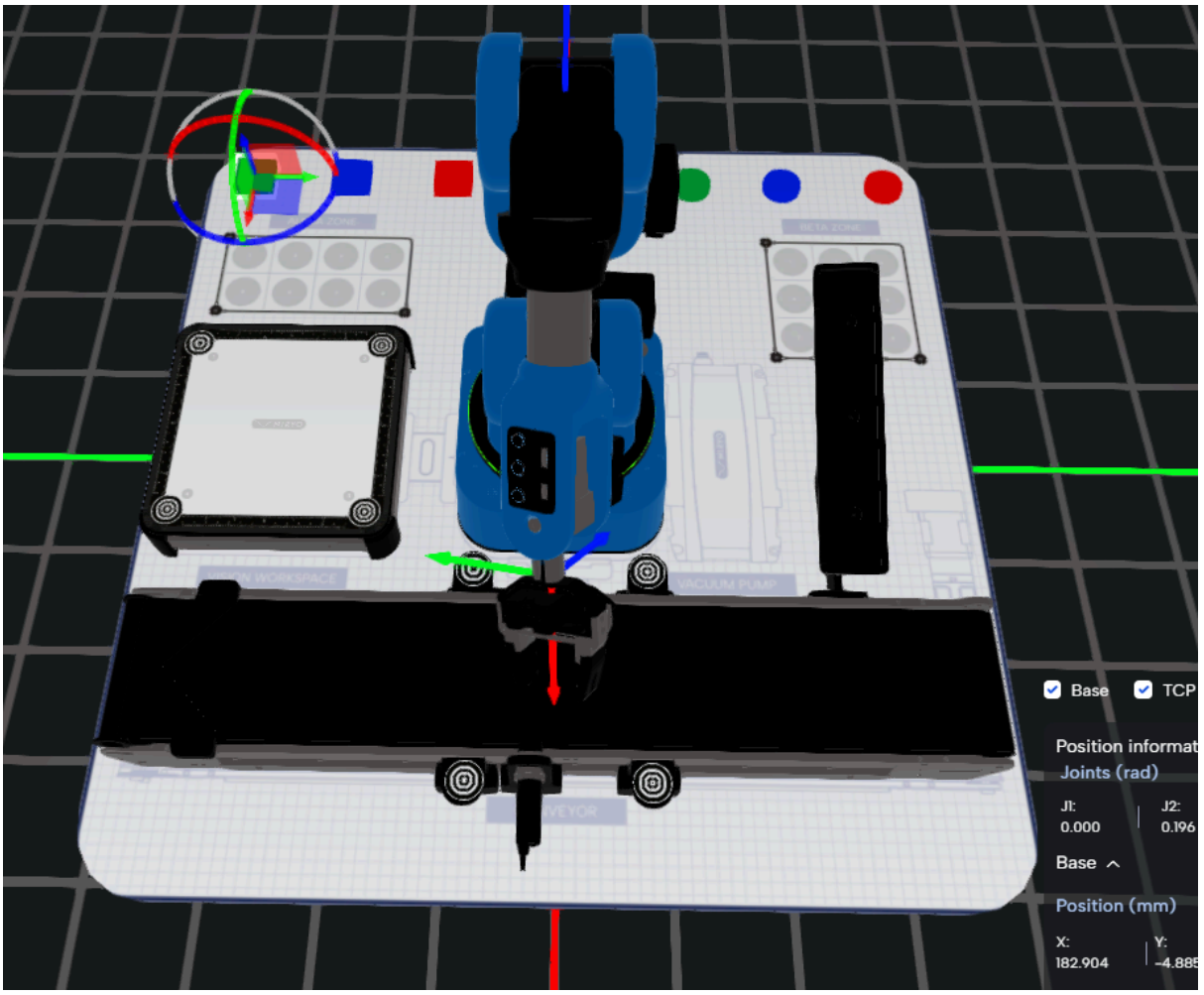
- You can adjust the size of the simulation by clicking and sliding the screen or clicking on the “Maximize” icon



Step 2: Quick Move on Top of Objects

- Double-click any object (workspace on the plate, workspace on the conveyor belt and tokens) in the simulation to move the robot above it.

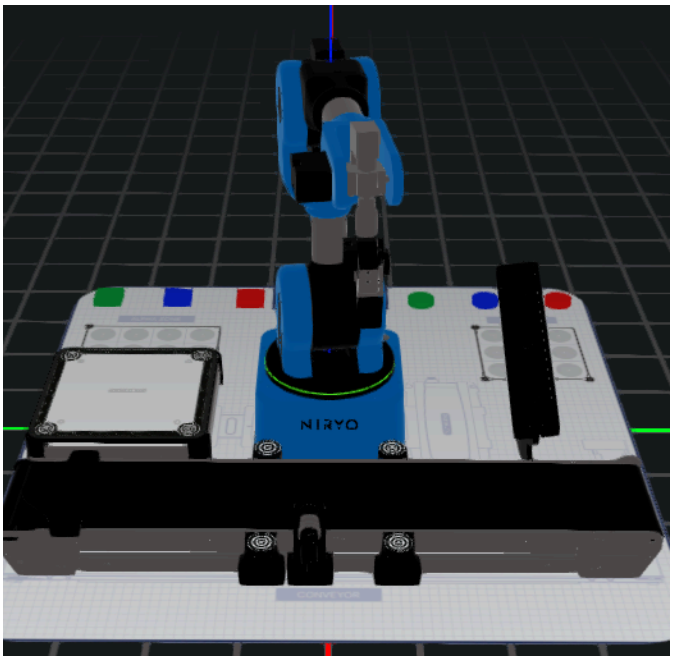
  Example for a square: click on it and move it in the scene with the gizmo



- If the object is a **vision workspace**, the robot automatically moves to a suitable **observation pose**.

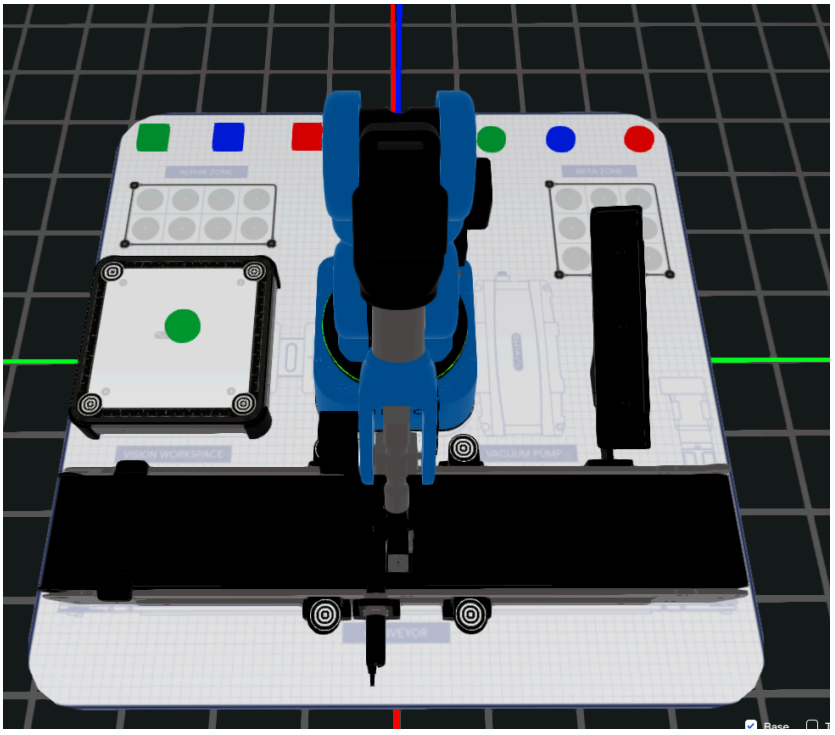
Step 3: Prepare the Scene

Set up the **conveyor workspace**, and the **IR sensor** as shown in the reference picture.You can drag the workspace and the IR sensor

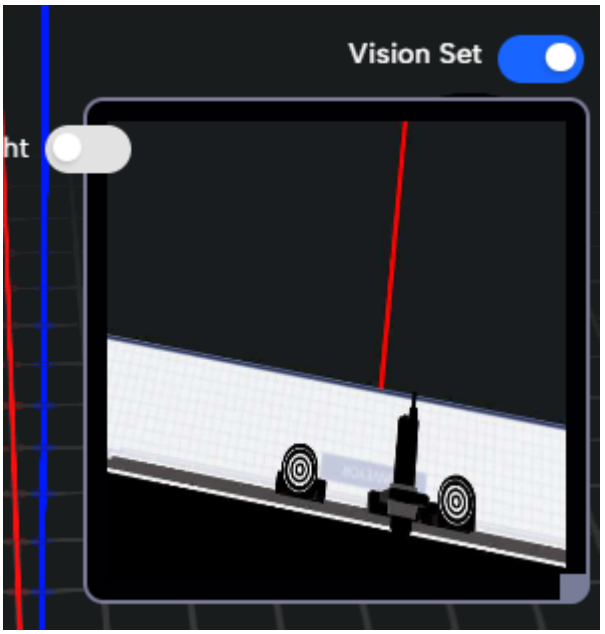


Step 4: Create Your First Program in Simulation

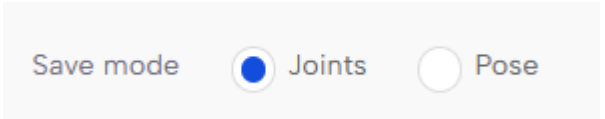
1. Open a new Blockly program in the “Programs”
2. Manually pick a circle from the storage area and place it on the **vision plate**.

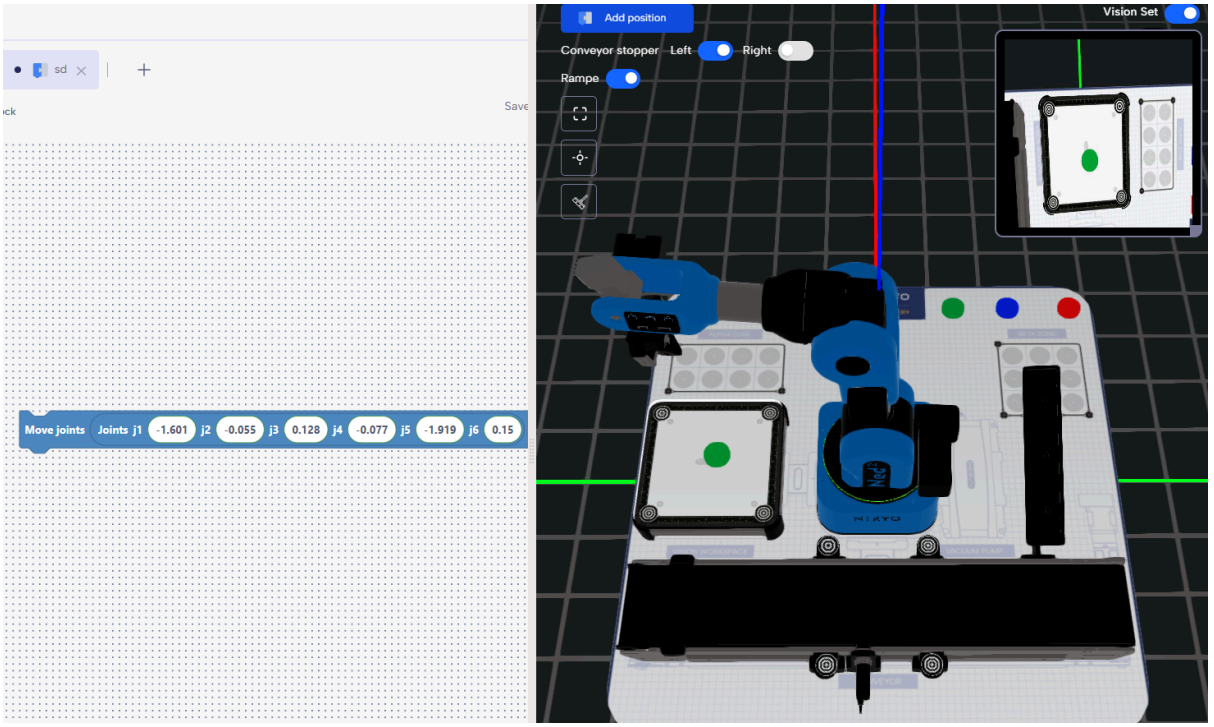


3. Open the camera feed by clicking **Vision Set** (top right).

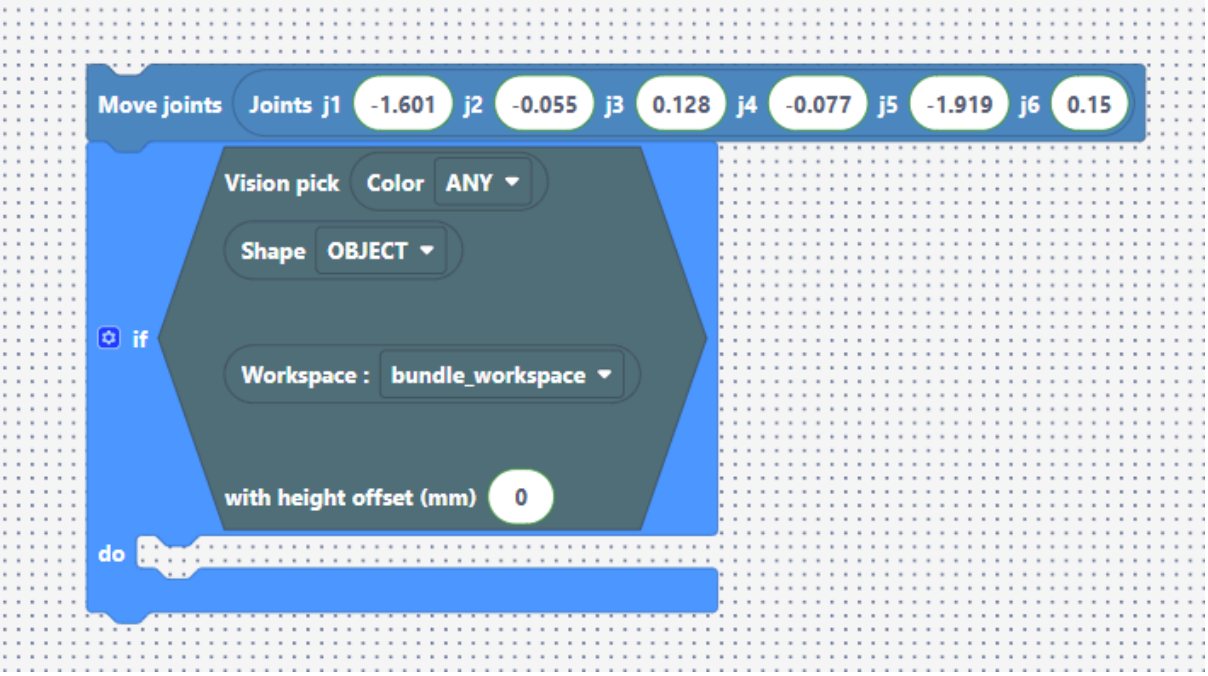


4. Double-click the vision plate and **save the current observation pose** using “Add position”.
   Advice: Be sure to use save mode ‘Joints’.

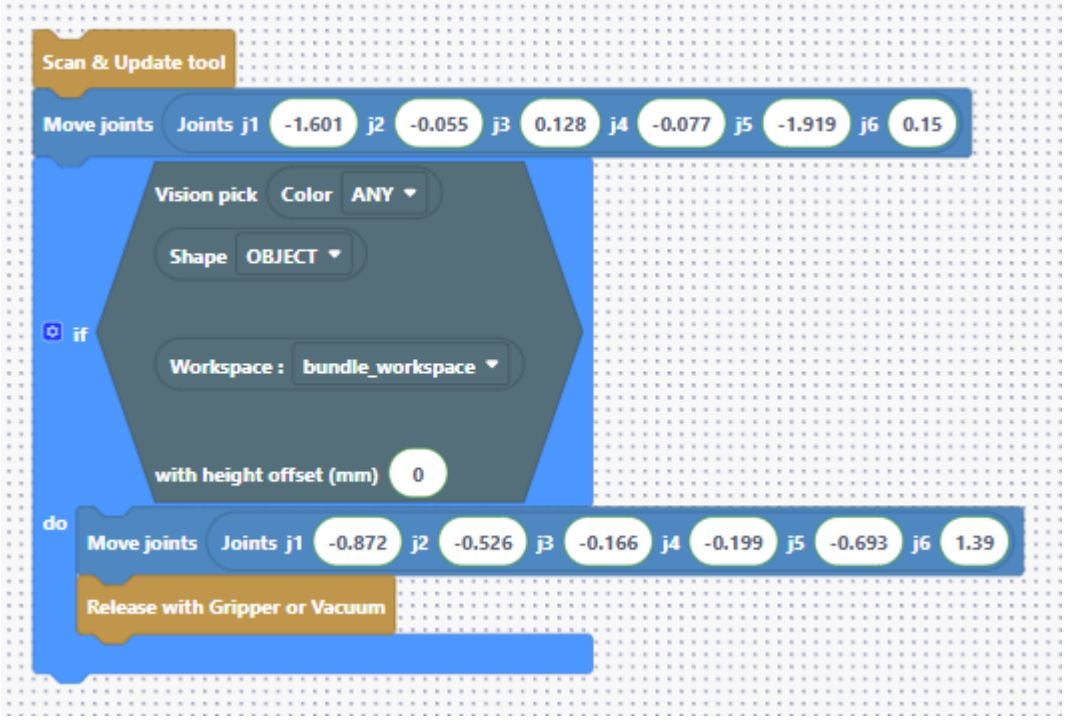




5. Add an **If block** (from the ***“Logic”*** section) and add the the **Vision Pick block** (from the ***“Vision”*** section), and select the correct workspace: `bundle_workspace`.



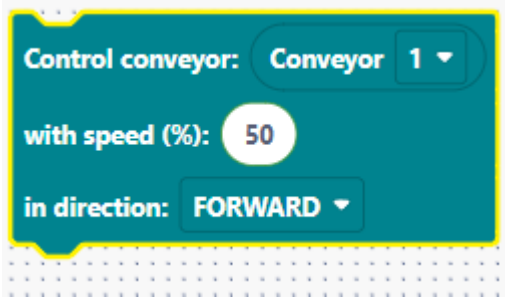
6. Move the robot above the **left side of the conveyor**, save this as your **place position**, and add a **Release with Tool** block. Put the 2 blocks in the “do” of the “if” block
    - Since you are using the gripper, add a **Scan Tool** block at the top of your program to ensure the robot detects the tool.



Add Conveyor Logic

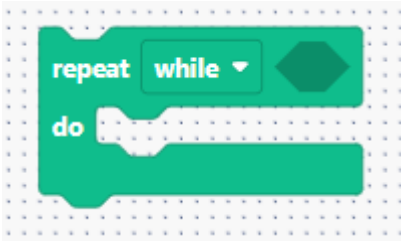
We want the conveyor to run **until the object enters the conveyor workspace**.

1. Add a **Run Conveyor** block. Put the **speed to 50%**.



Since you are using the conveyor, add a **Scan conveyor** block at the top of your program to ensure the robot detects the conveyor.

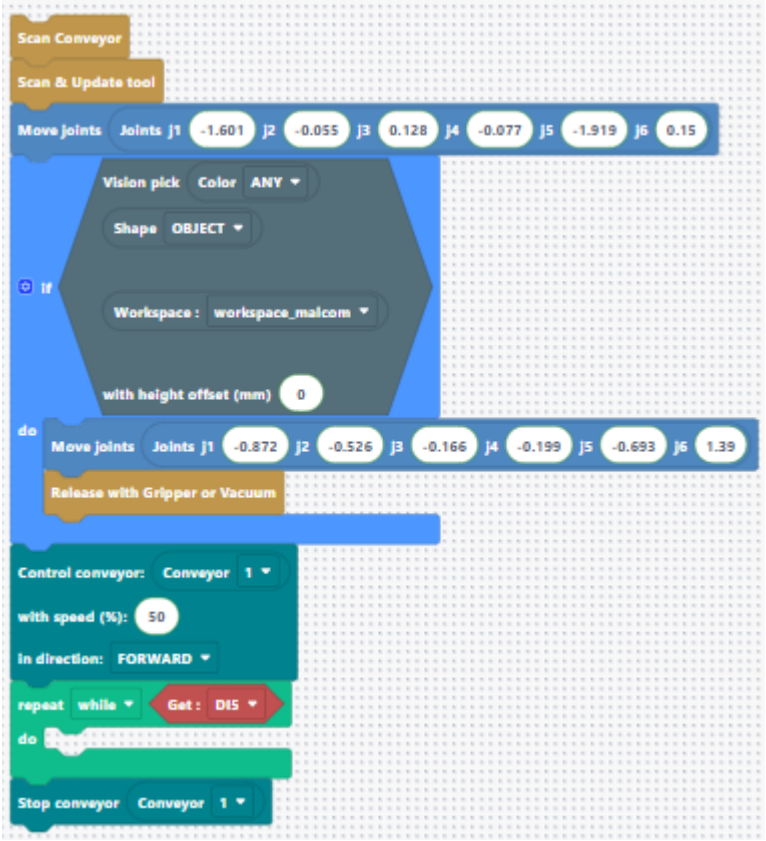
2. Add the logic for the IR sensor:
    - Reminder: the IR sensor is plugged into **DI5**.
        - **DI5 = HIGH (True)** → no object detected
        - **DI5 = LOW (False)** → object detected
3. Add a ***Repeat While*** block:



and add the condition that triggers the break out of the loop : 

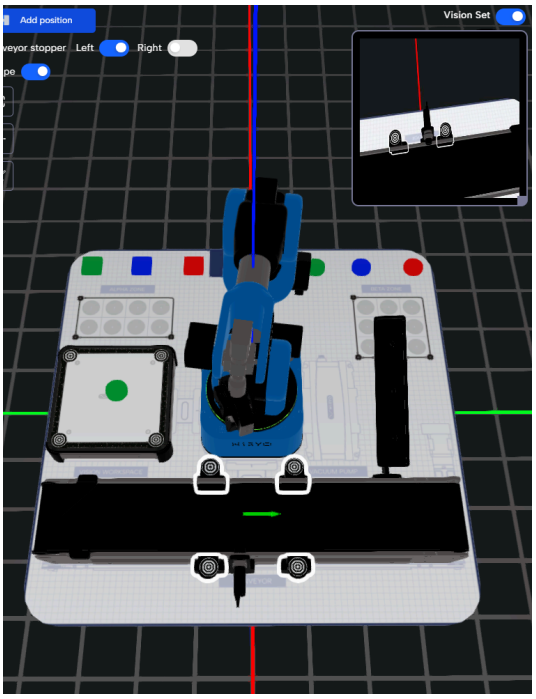
    - `While Get DI5` → do nothing (wait).

4. After the While, the stopping condition is reached when the object moves in from of the IR sensor. The robot breaks out of the loop and we **stop the conveyor**.

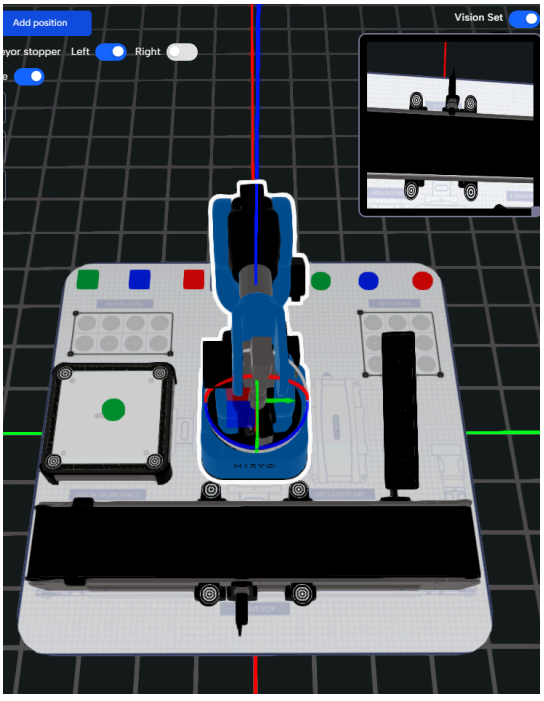


Vision Pick on the Conveyor Workspace

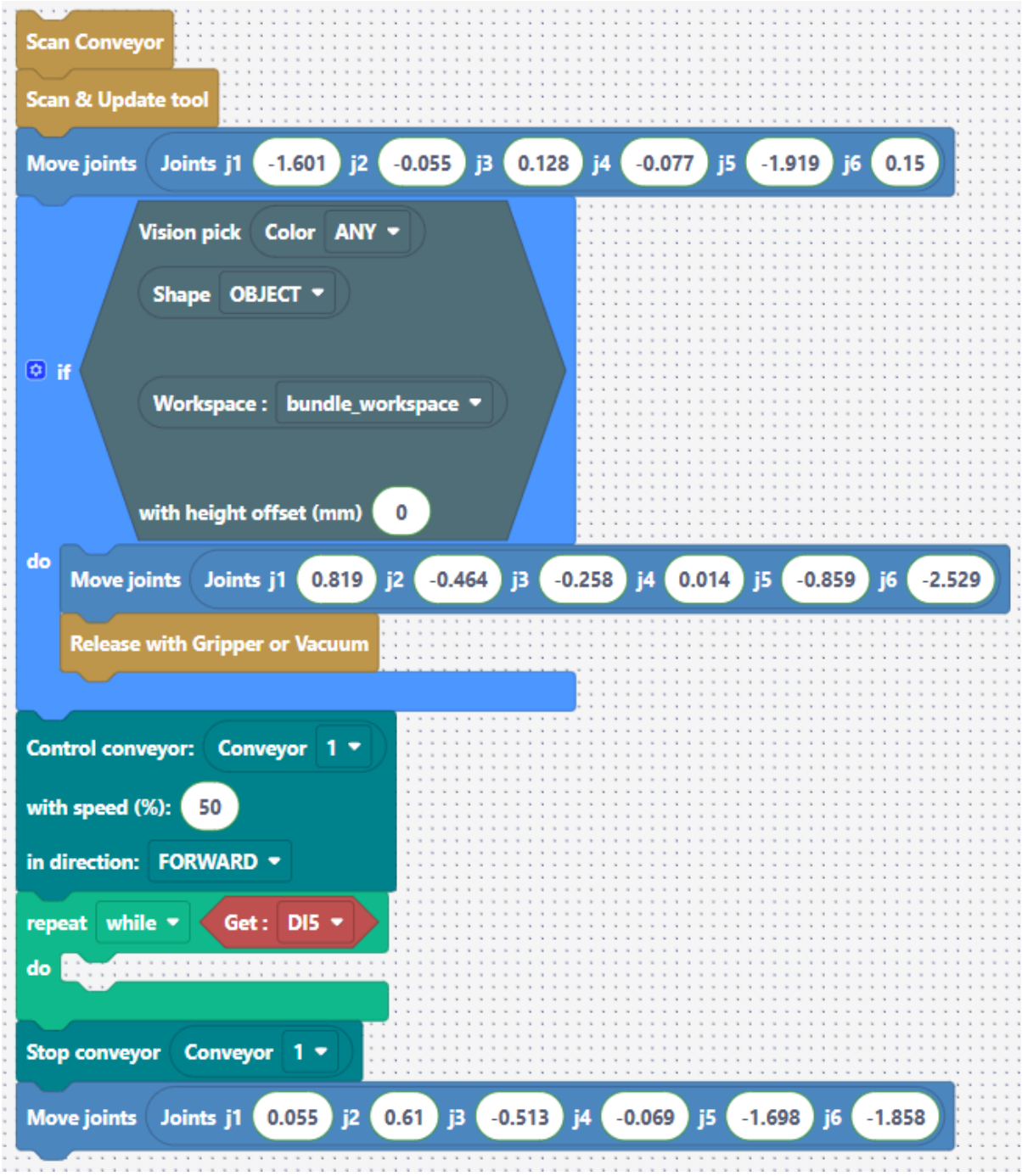
1. Move the robot to the correct observation pose for the **conveyor workspace**.
    - Warning: the conveyor workspace is movable, so the double-click may not be accurate.



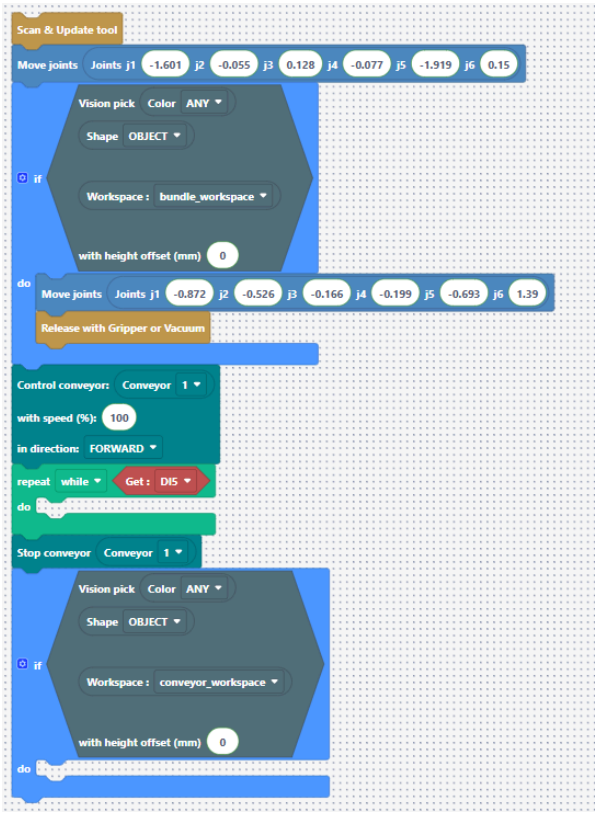
- Adjust the pose manually if needed.



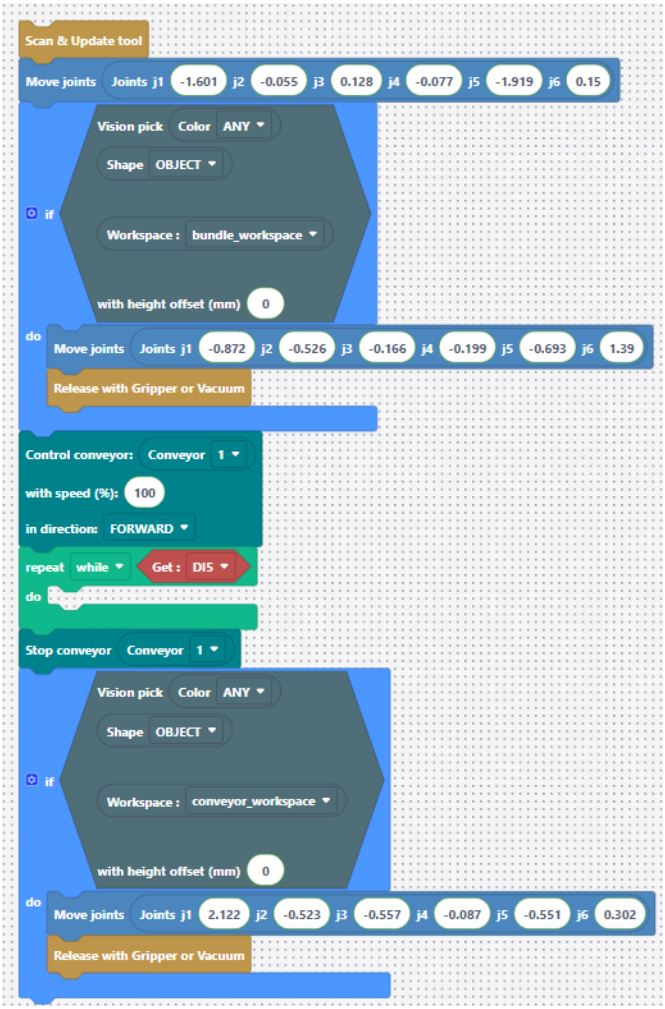
- Save the observation pose using the “Add position” button.



2. Create a **Vision Pick** exactly as you did for the vision plate, but using the **conveyor workspace**.


Scan & Update tool
Move joints Joints j1 -1.601 j2 -0.055 j3 0.128 j4 -0.077 j5 -1.919 j6 0.15
if
Vision pick Color ANY
Shape OBJECT
Workspace : bundle_workspace
with height offset (mm) 0
do
Move joints Joints j1 -0.872 j2 -0.526 j3 -0.166 j4 -0.199 j5 -0.693 j6 1.39
Release with Gripper or Vacuum
Control conveyor: Conveyor 1
with speed (%): 100
in direction: FORWARD
repeat while Get : DI5
do
Stop conveyor Conveyor 1
if
Vision pick Color ANY
Shape OBJECT
Workspace : conveyor_workspace
with height offset (mm) 0
do

3. Move the robot manually above the **Beta zone frame**, **save** this position, and add a **Release with Tool** block.

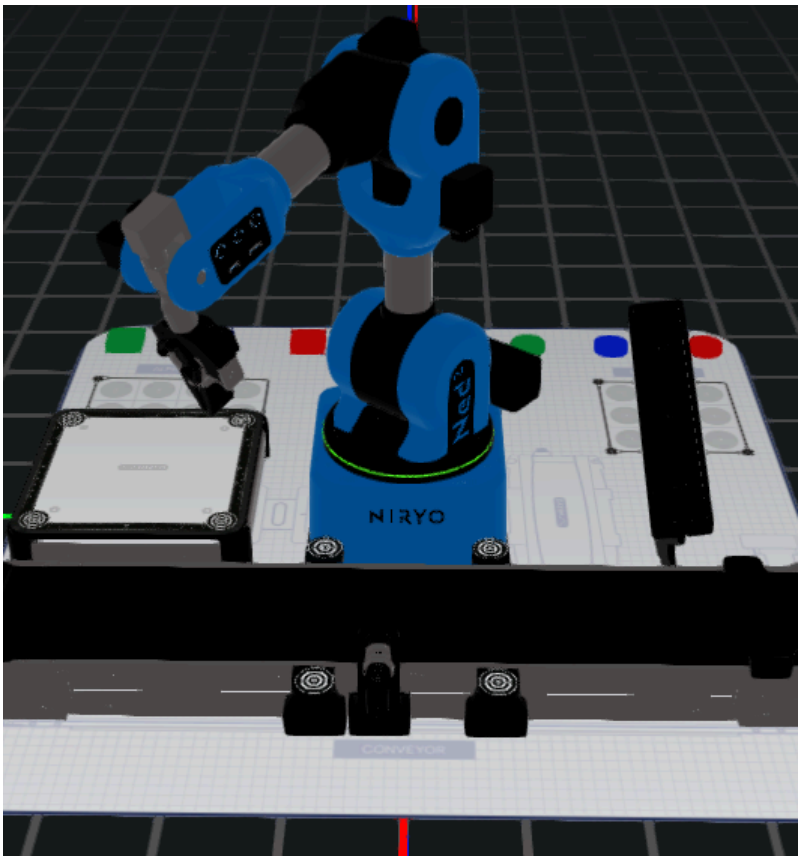


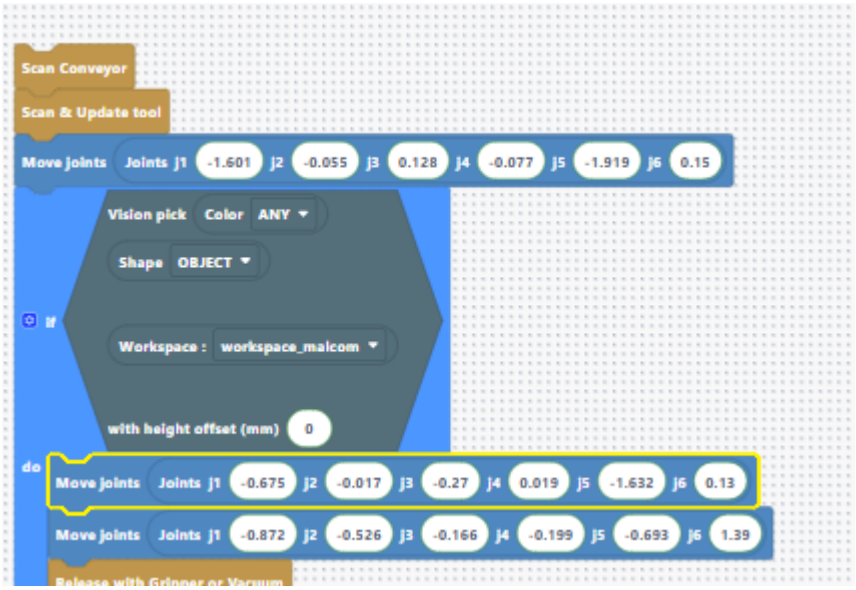
Step 5: Improve the Code Before Switching to the Real Robot

You can now run the program and observe the simulation.
The program works — but you may notice the robot passes **through obstacles** like the conveyor slope.

In the real world, this is not acceptable.
To safely transfer the program to the physical robot, we must add **intermediate positions**:

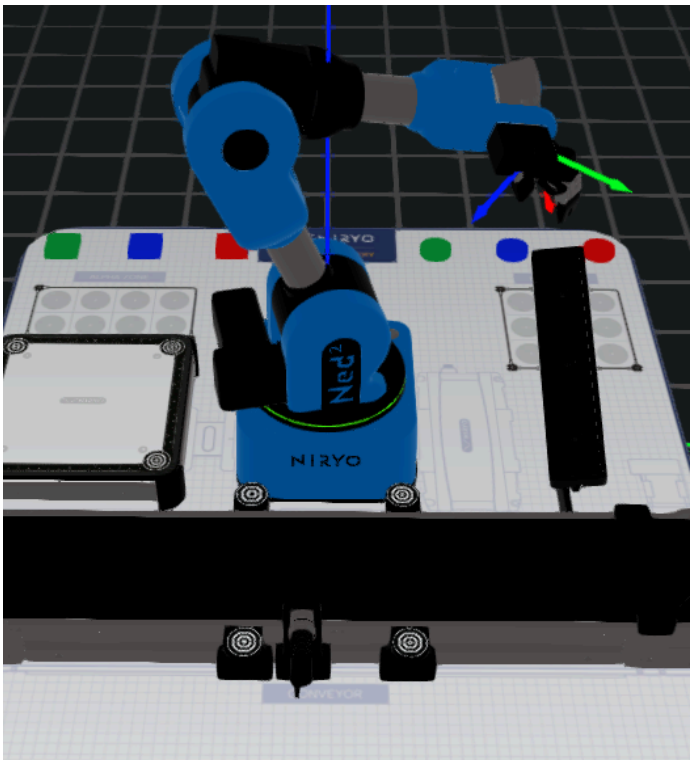
1. **After the Vision Pick on the Vision Plate:**
    - Move to a position between the vision plate and the conveyor to ensure clearance before moving toward the conveyor.

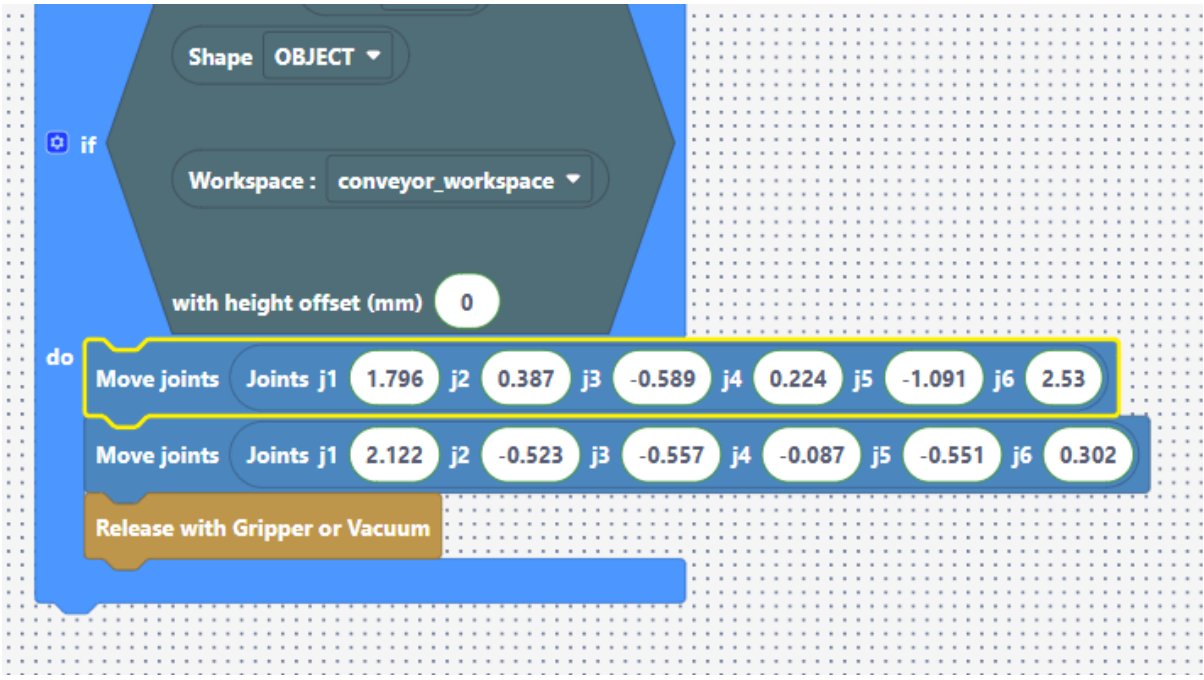




2. **After the Vision Pick on the Conveyor:**

    - Move to a position **high above the conveyor** before traveling toward the Beta zone placement position.





Step 6: Transition to the Physical World

Action:

If your setup is not already prepared, do the following:

- Mount the **bundle** as shown in the reference picture.
- Connect to the **real robot** and perform a **calibration**.
- Create the workspaces for the **vision plate** and the **conveyor**.
    - To match the simulation, name them:
        - `bundle_workspace` (vision plate)
        - `conveyor_workspace` (conveyor)

Disconnect from the simulation and reconnect to the real robot.

Since the simulation and physical setup may not match perfectly, the **observation poses** may be inaccurate in the real world.
Run your program and observe the movements. If a pose needs correction:

- Open a **new program**,
- Copy-paste your Blockly code,
- Adjust only the positions in the new program (to avoid breaking the simulation version).

Finally, place an object on the **vision plate** and run the program.

Result:
✅ Your program built in simulation now works with the real robot!