# EXERCICES

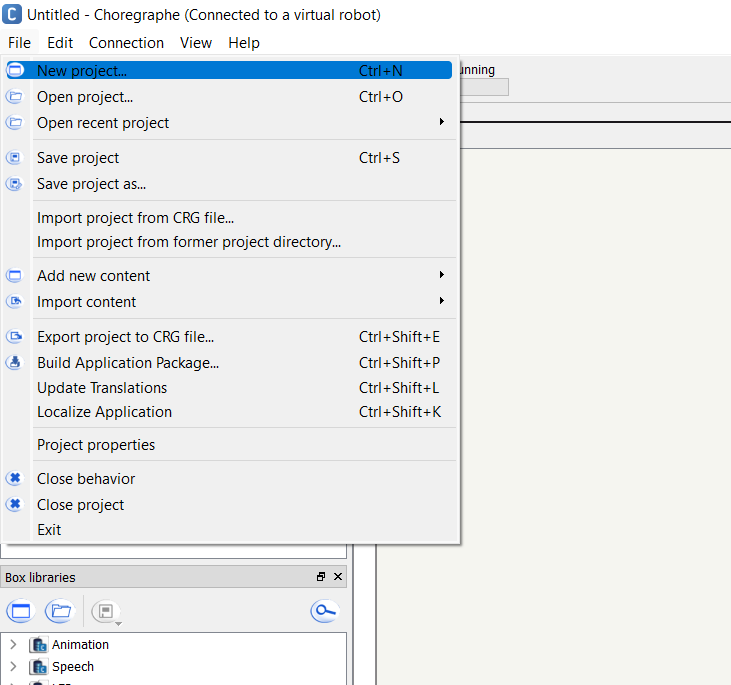
## : Making Pepper Speak

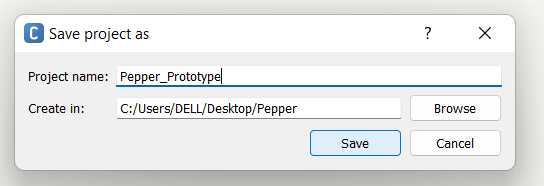
**Objective:**

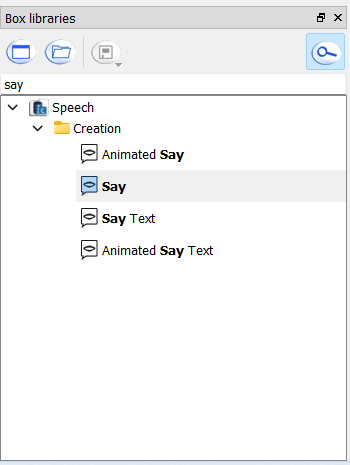
Learn how to make Pepper say a simple phrase using Choregraphe.

**Steps:**

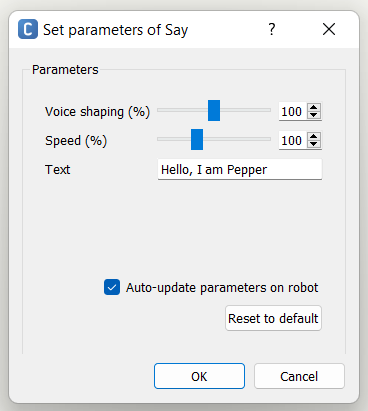
1. Open Choregraphe and create a new project (File > New Project).



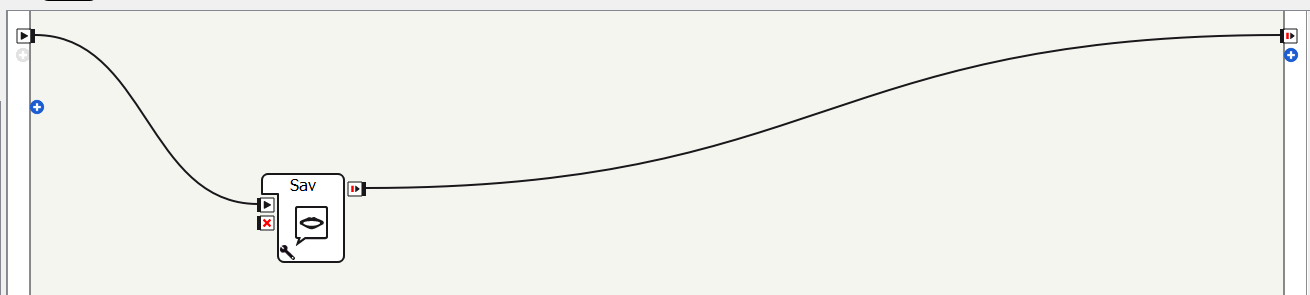
1. Enter a simple name, for example, “Pepper\_Prototype.”
2. Save the project in a dedicated folder. 
3. Locate the **Say Box** in the Box Libraries Panel.



1. Drag and drop the Say Box into the Flow Diagram Panel.
2. Double-click the Say Box and enter the text: *"Hello! I am Pepper."*



1. Connect the **onStart** input to the Say Box and connect the **onStopped** output to the Stop Box.



1. Click **Play** to test Pepper’s speech in simulation.
2. Deploy it to Pepper and observe the result.

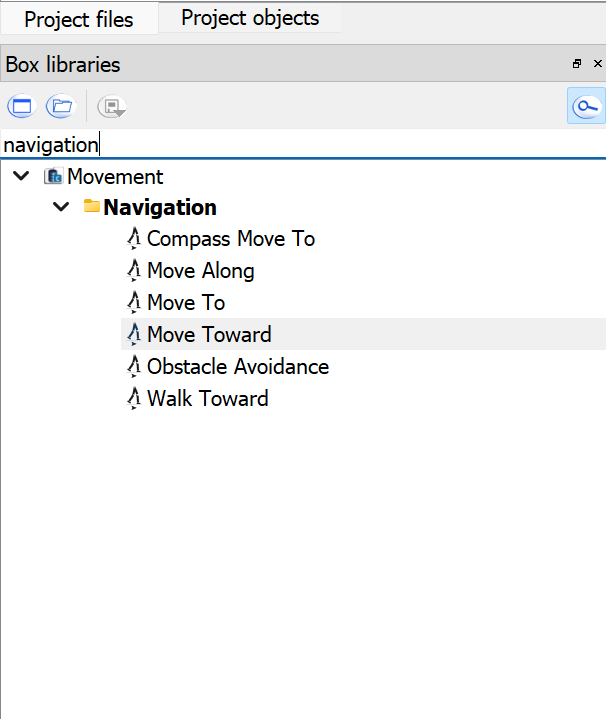
## : Making Pepper Move

**Objective:**

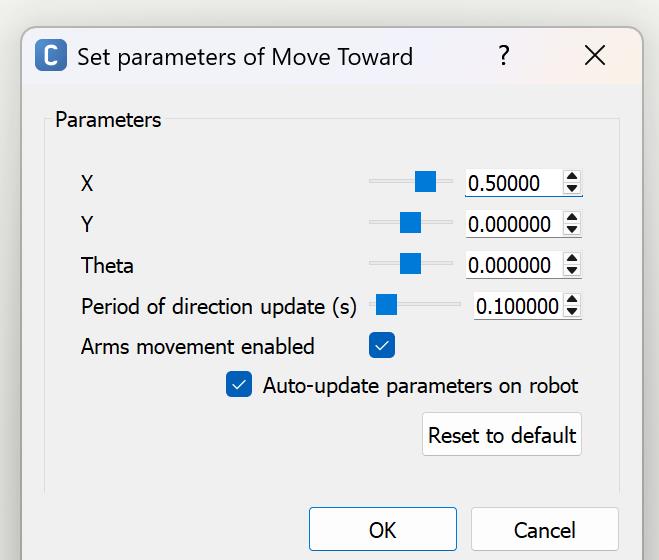
Learn how to program Pepper to perform basic movements.

**Steps:**

1. Open a new project in Choregraphe.
2. Find the **Move Toward Box** in the Box Libraries Panel.



1. Drag it into the Flow Diagram Panel.
2. Set movement parameters (e.g., move forward 0.5 meters).



1. Connect the Start Box to the Move Toward Box.



1. Click **Play** and test the movement on Pepper.

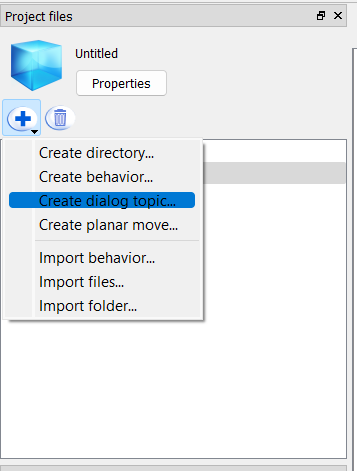
## : Simple Interaction with Pepper

**Objective:**

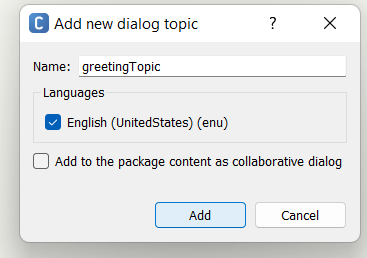
Create a basic interactive conversation using **Dialog Boxes**.

**Steps:**

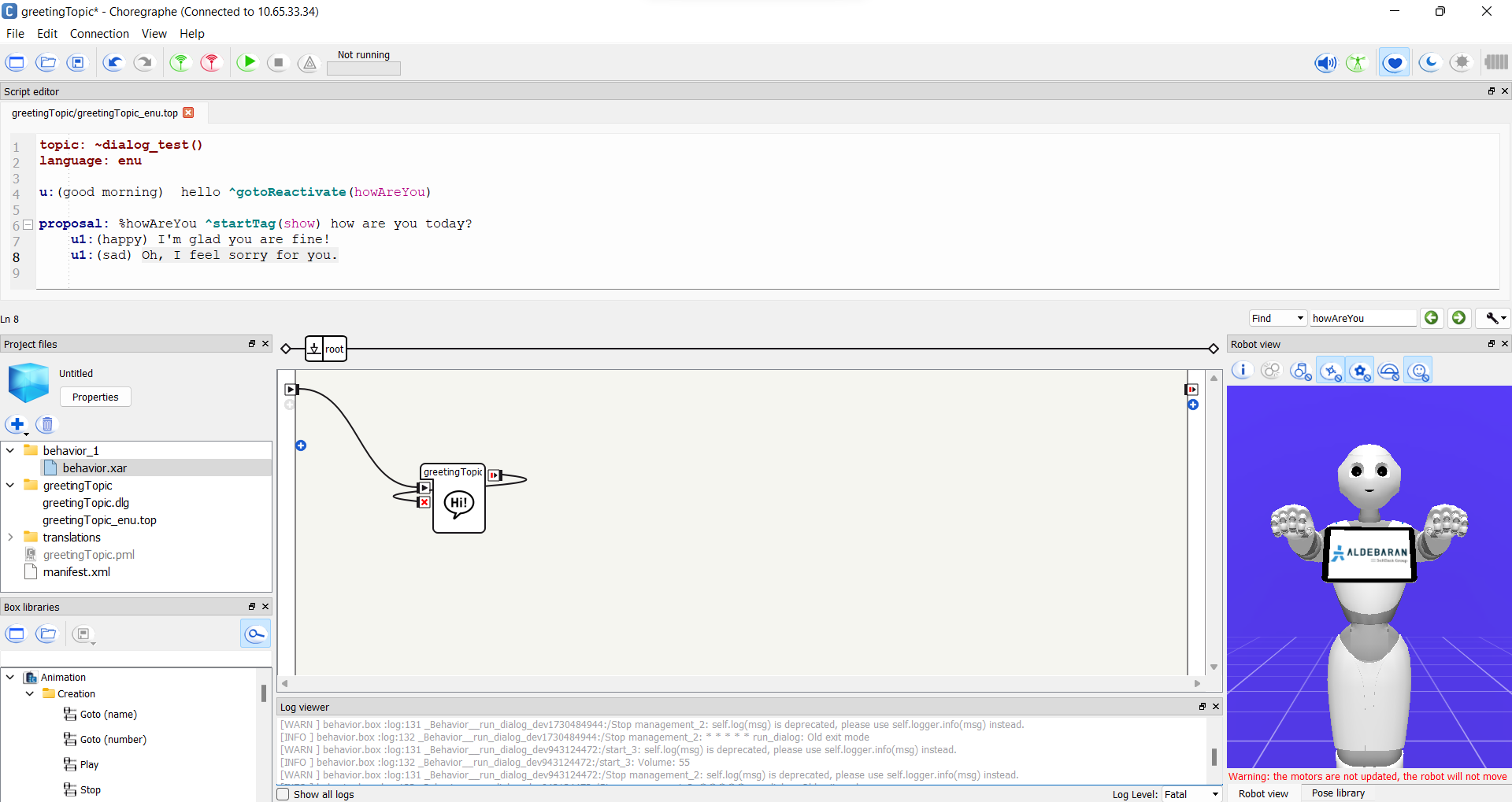
1. Open Choregraphe and go to the dialog section.
2. Click on “Create dialog topic.”



1. Enter a topic name without spaces (for example, “greetingTopic”).
2. Select the language (e.g., English) and click Add. This generates two files: a .dlg file (the dialog structure) and a .top file (the rules and tags).



1. Locate the generated .dlg file in the project files. This file holds the basic structure of the dialog. Also, find the .top file. This file is where you define the detailed rules for how Pepper will respond.
2. Open the .dlg file in built-in editor. This file might contain a basic template. You can leave it as is for now, as most of your detailed rules will be in the .top file.
3. Open the .top file for your “greetingTopic”. This is where you define the rules for Pepper’s responses.
4. Now, we will define a basic greeting rule by write a rule that listens for a greeting using qiChat. “u: (Hello) Hi there! How can I help you?” means whenever user says “Hello”, pepper will respond with ‘Hi there! How can I help you?’. Once .top file is edited, save the changes.
5. Add the dialog box to the behavior flow.
6. Click Play to test the dialog.

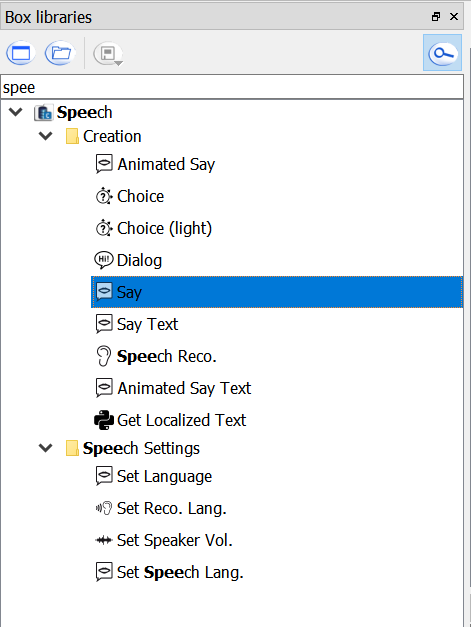


## : Making Pepper Speak with Different Voices

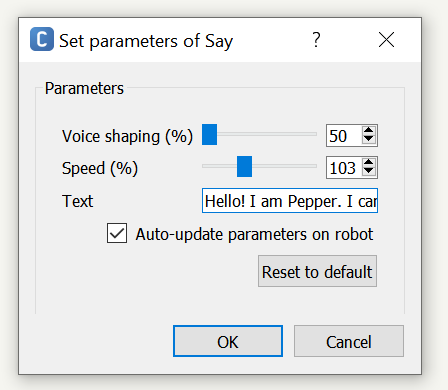
**Objective:**  
Learn how to change Pepper’s voice settings while making it speak.

**Steps:**

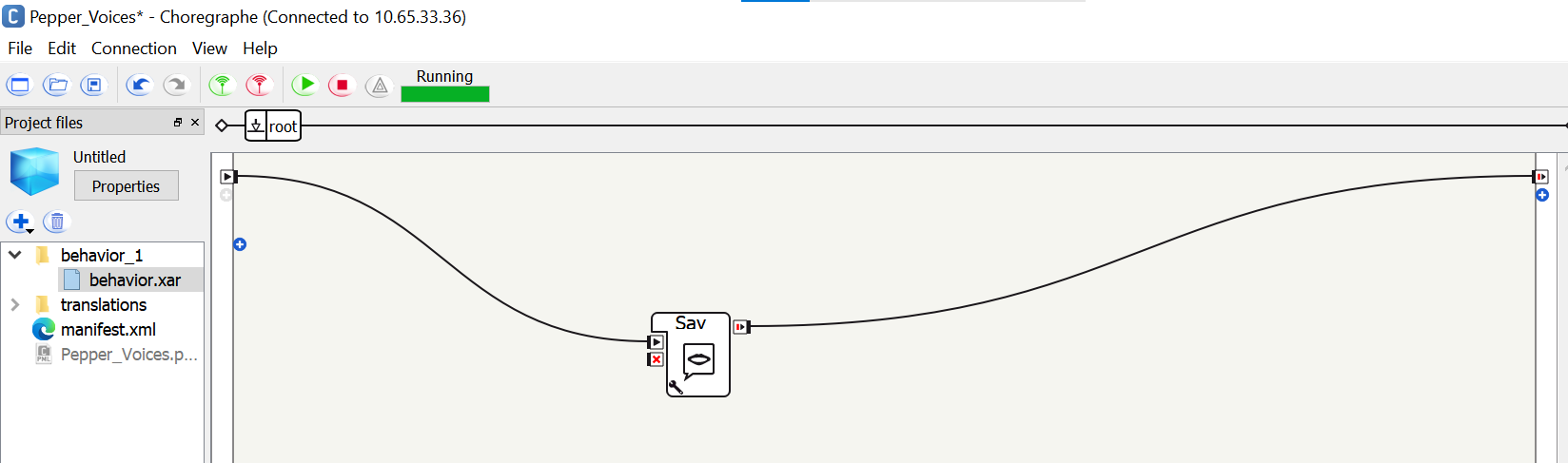
1. Open Choregraphe and create a new project (File > New Project).
2. Name it “Pepper\_Voices.”
3. Locate the **Say Box** in the **Box Libraries Panel** under Speech and drag it into the **Flow Diagram Panel**.



1. Right-click the Say box and click ‘Set Parameter’. Enter the text: "Hello! I am Pepper. I can change my voice."
2. Adjust the speed and voice shaping and test with different values.



1. Connect the **onStart** input to the **Say Box** and connect the **onStopped** output to the **Stop Box**.



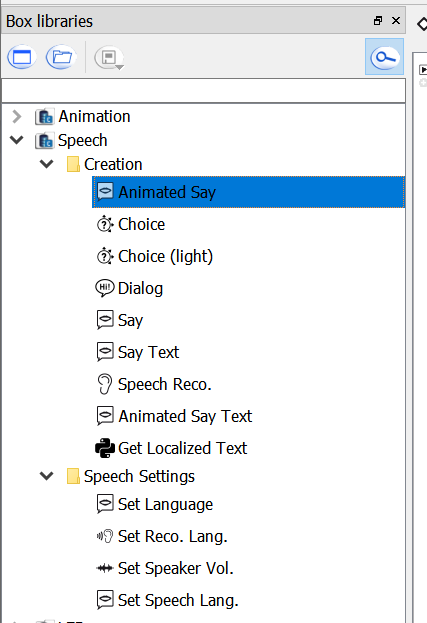
1. Click **Play** to test the voice in simulation.
2. Deploy it to Pepper and listen to the difference in voice settings.

## : Making Pepper Perform a Wave and Speak

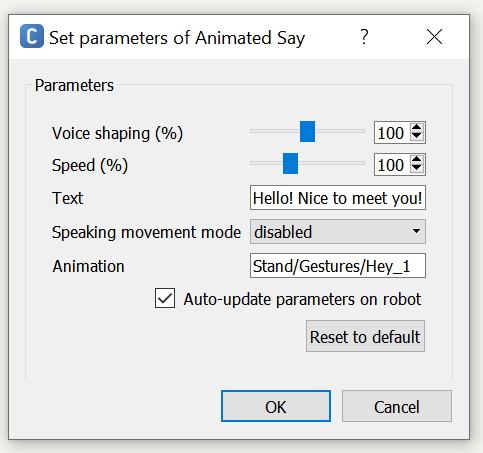
**Objective:**  
Learn how to make Pepper combine gestures and speech.

**Steps:**

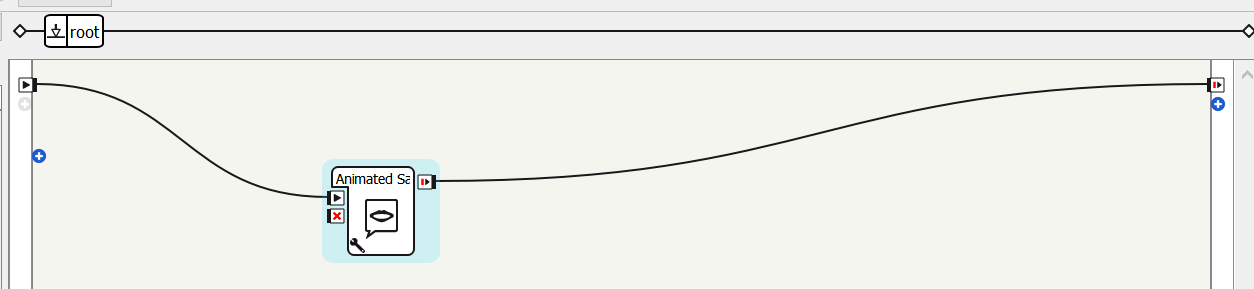
1. Open Choregraphe and create a new project named **“Pepper\_Wave”**.
2. Locate the **Animated Say Box** in the **Box Libraries Panel**.



1. Drag it into the **Flow Diagram Panel**.
2. Right-click and open ‘Set Parameters’ and enter the text: "Hello! Nice to meet you!"
3. Set animation to ‘Stand/Gestures/Hey\_1’ which is the wave animation of the pepper.



1. Connect the **onStart** input to the **Wave Box**, then connect the **onStopped** output of the **Wave Box** to the **Animated Say Box**.



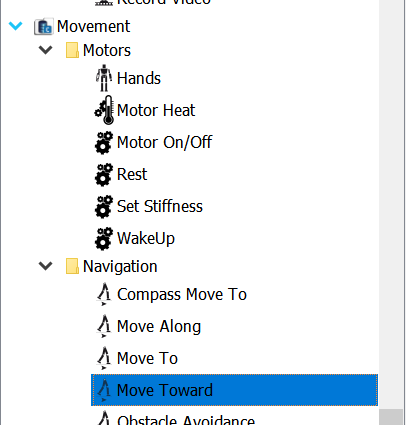
1. Click **Play** to test the movement in simulation.
2. Deploy it to Pepper and observe Pepper’s hand wave and speech.

## Exercise 7: Make Pepper Spin

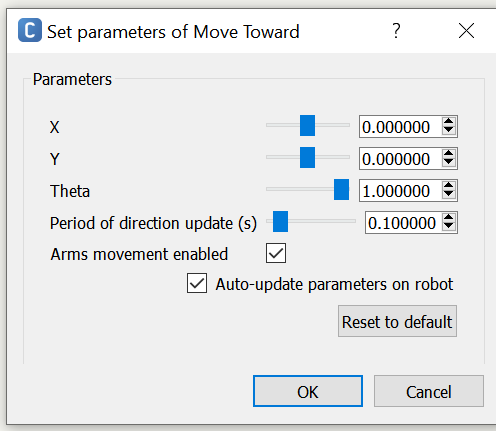
**Objective:**  
Learn how to program Pepper to move in a specific pattern.

**Steps:**

1. Open Choregraphe and create a new project called **“Pepper\_Spin**
2. Locate the **Move Toward Box** in the **Box Libraries Panel**.



1. Set Theta to 1 which helps pepper to rotate in radians. And click OK.



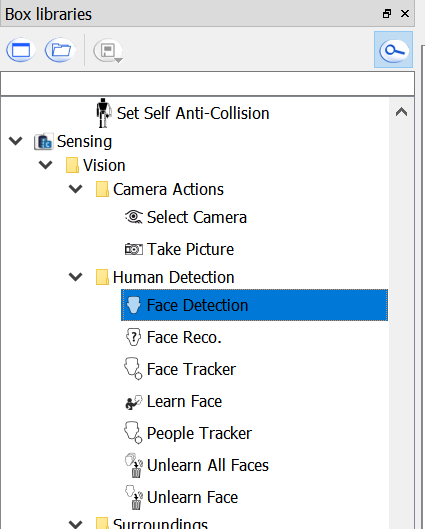
1. Connect the **onStart** input to the **Move Towards Box**, then connect the **onStopped** output of the **Wave Box** to the **Move Towards Box**.
2. Click **Play** to test the movement in simulation.
3. Deploy it to Pepper and observe its spinning movement.
4. Click stop in order to stop the behavior.

## Exercise 8: Making Pepper Detect and Respond to a Face

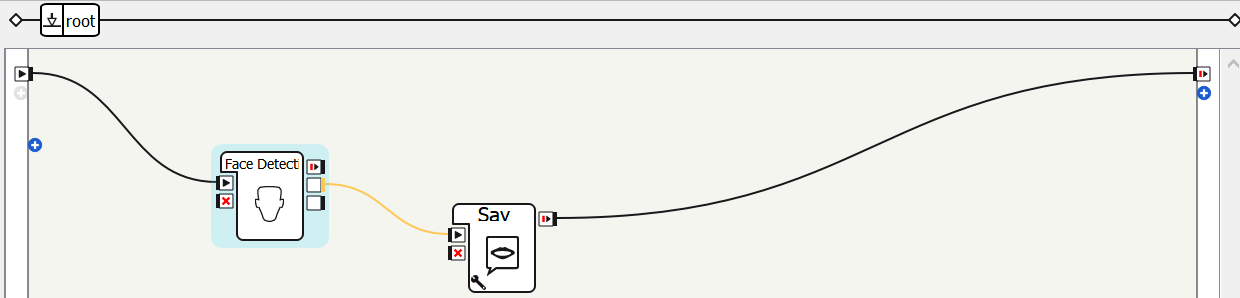
**Objective:**  
Learn how to make Pepper recognize a human face and respond.

**Steps:**

1. Open Choregraphe and create a new project called **“Pepper\_FaceDetection.”**
2. Locate the **Face Detection Box** in the **Box Libraries Panel**.



1. Drag it into the **Flow Diagram Panel**.
2. Drag a **Say Box** and enter the text in the parameter: "I see someone!"
3. Connect the **Face Detection Box**’s no of faces output to the input of the **Say Box**.



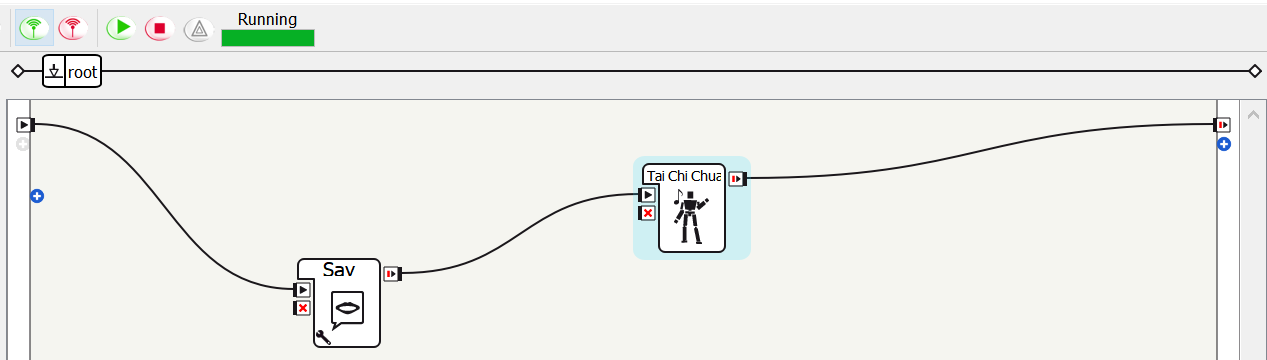
1. Click **Play** to test in simulation.
2. Deploy it to Pepper and see if it recognizes a face and responds.

## Exercise 9: Making Pepper Dance

**Objective:**  
Learn how to make Pepper perform a relaxing Tai Chi Chuan dance using the built-in behavior box in Choregraphe.

**Steps:**

1. **Open Choregraphe** and create a new project called ‘TaiChi\_PepperDance’.
2. In the **Box Libraries Panel** search for the box named **“Tai Chi Chuan”** under animation.
3. **Drag and drop** the **Tai Chi Chuan** box into your **Flow Diagram Panel**.
4. Also locate and drag a **“Say”** box from Box Libraries.
5. Right-click the Say box and open ‘set parameter’.
6. Enter the phrase into text " I will show you my Tai Chi Chuan dance!"
7. Connect the **onStart output** of the Start box to the Say box.
8. Connect the output of the Say box to the input of the Tai Chi Chuan box.
9. Then, connect the output of Tai Chi Chuan to the Stop box.
10. Your flow should look like this:



1. Click **Play** to test the dance in simulation mode. You’ll see Pepper performing a graceful Tai Chi sequence.

## Exercise 10: Making Pepper Follow a Person

**Objective:**  
Learn how to use Pepper’s tracking abilities.

**Steps:**

1. Open Choregraphe and create a new project called **“Pepper\_Follow”**.
2. Locate the Python Script in the **Box Libraries Panel**.
3. Drag **the python script** into the **Flow Diagram Panel**.
4. Edir script and paste the following code in it:

from naoqi import ALProxy

class MyClass(GeneratedClass):

def \_\_init\_\_(self):

GeneratedClass.\_\_init\_\_(self)

self.PEPPER\_IP = "10.65.33.36"

self.PORT = 9559

self.tracker = None

self.motion = None

self.tts = None

def onLoad(self):

try:

self.tracker = ALProxy("ALTracker", self.PEPPER\_IP, self.PORT)

self.motion = ALProxy("ALMotion", self.PEPPER\_IP, self.PORT)

self.tts = ALProxy("ALTextToSpeech", self.PEPPER\_IP, self.PORT)

except Exception as e:

print("Error during proxy creation:", e)

def onUnload(self):

if self.tracker:

try:

self.tracker.stopTracker()

self.tracker.unregisterAllTargets()

except Exception as e:

print("Error during tracker cleanup:", e)

def onInput\_onStart(self):

try:

if self.tts:

self.tts.say("I will now start following you.")

if self.tracker:

self.tracker.registerTarget("People", 0.5)

self.tracker.setMode("Move")

self.tracker.track("People")

except Exception as e:

print("Error during tracking start:", e)

# self.onStopped() # Uncomment if needed to continue flow after start

def onInput\_onStop(self):

self.onUnload()

self.onStopped()

The onload() function connects to the al modules. ALTracker allows Pepper to track and follow objects. ALMotion controls Pepper's movements like walking, turning, etc. ‘self.tts.say("I will now start following you.")'. self.tracker.registerTarget("People", 0.5) tells Pepper to track people. ‘self.tracker.setMode("Move")’ sets Pepper's tracking mode to **Move**, so it will actually **walk and follow** the person and ‘self.tracker.track("People")’ starts the tracking behavior for the target "People".

1. Connect the **onStart** input to the Python ScriptBox to the start box, then connect the **onStopped** of the python script box to the end.
2. Click **Play** to run the behavior of pepper.
3. Pepper should say: "I will now start following you."
4. Then it will start tracking and following any nearby person it detects.
5. Click the "Stop" button in Choregraph to stop the behavior.