Week 4 Lab Portfolio Sensors

Thawatchai Sangthep , P2681054 , MSc Intelligent systems

Introduction

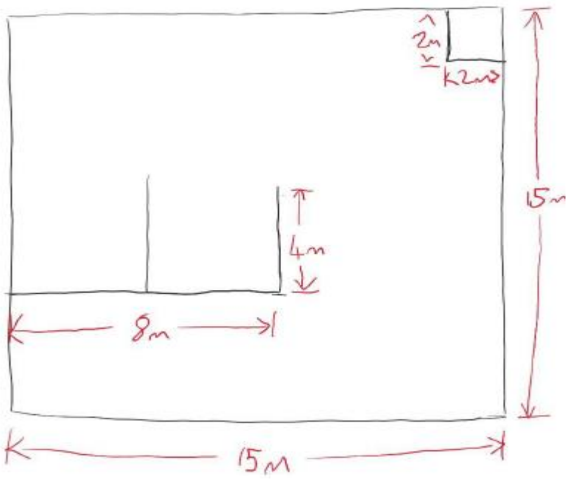
According to robot is a tool to do work by using program. It has a different design that it would be anything what the engineer wants such as wheel or arm that depends on the environment and mission. The people usually use robot to help own work such as explorer, factory, medical and etc. By the ways, this assignment has a target to make a robot by using programming to run follow the wall. The task seems basic, but it must use many skills to do complete. Firstly, strategy, it must design the way to run of robot and after detected the wall what the robot should do later. Secondly, it has to study and predict about the robot can be that it must use logical to do. Thirdly, community such as talk with each other about the task, what should to do, step of thinking to do, the using tools. Lastly, make to understand about coding what the line is mean and how to write the code.

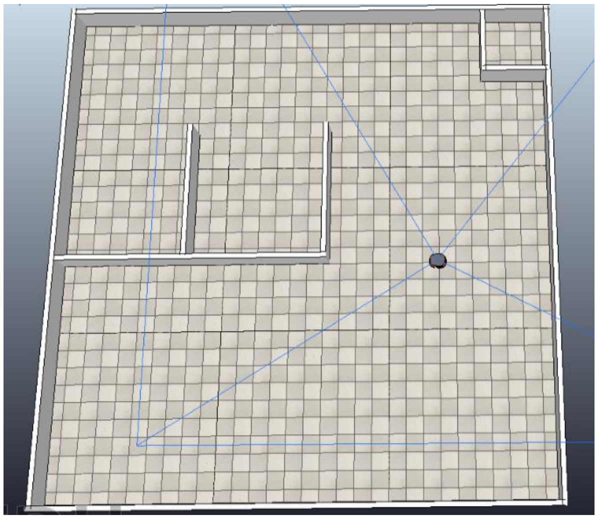
The task

Due to an assignment has a target to set a robot run randomly in the environment if the robot sees a wall then it should follow the wall in a determined direction. It has many conditions to do. Firstly, consider the random distance to travel in a straight line. Secondly, consider the random angle to turn to and which random direction. Thirdly, In V-REP cannot tell the robot how far to travel or how much to turn by so you will need to devise a way to allow the robot to travel randomly. So, the task has many details to do that I have to study and make to clearly understand before start.

Previous knowledge and statements

1. The map has a wall which is the skeleton of the path to follow.
2. It has some lines that indicate the ideal path that the robot should follow.
3. Create a bounding box around the area so the robot cannot escape.





Which is achieved through a literature review or suitable background section

The map is finish that it has an environment that use the wall the way to exit and cannot move when the robot impact. I put some feature such a felt pen to see the track when the robot moves.

What did I do ?

Due to the task has a mission to order the robot move follow the wall but have many mini conditions such as consider the random distance to travel in a straight line., consider the random angle to turn to and which random direction., In V-REP cannot tell the robot how far to travel or how much to turn by so you will need to devise a way to allow the robot to travel randomly. So, it must design and study to use a suitable to command the robot such as, strategy and language of programming

Firstly, strategy, I have many questions to think about it because I do not have experience with robot or coding to control it. For example, the basically how to make a robot move, how to make a robot turn, how to use the sensor, how many sensors that robot uses and when the robot detected the wall what should it do.

I solve a problem by searching on internet about the basically, how to control the robot. Something I got an answer from webpage, another way is on YouTube that it is an important website because it shows information that it is necessary such as how to do, why it be and combine with example. If some topic I doubt and I cannot find the answer, I will ask with classmate who would love to help.

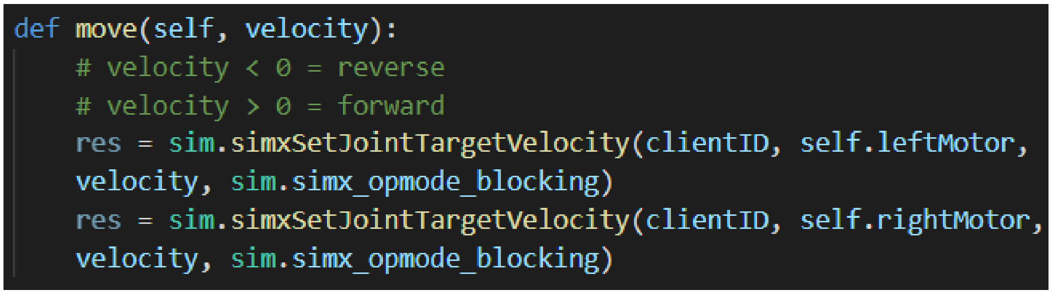
Secondly, language of programming, I am using Python language on version 3.10.0 because it is up to date. Next reason is a computer programing language often used to build website and software automate tasks. Plus, it is easy to understand and more classmate use. But I have a problem with programming skill because I do not have experience develop by using python.

In past, I decided use Matlab because I am familiar. But next week later, I knew that, on the class does not have someone use Matlab. So, I consider moving to use python language because I am not professional on Matlab if I have some issues, I will not have other to consult. Next reason, it can find example easier than Matlab and I can ask with classmate or someone on internet to help me because it has wide community when I stuck in trouble.

What have I learned by doing this lab

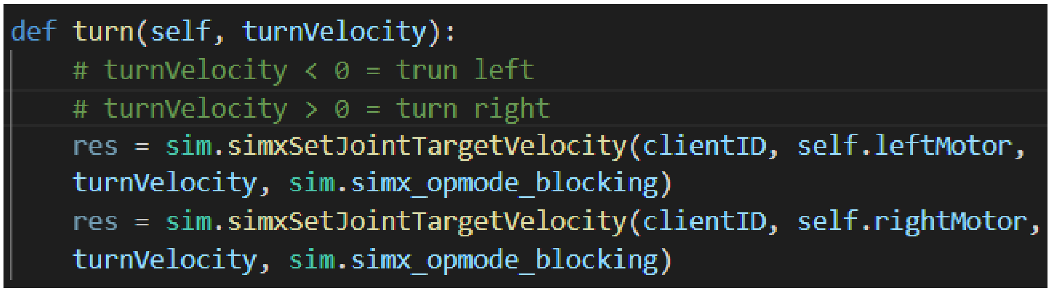
I have improved many skills and knowledges to learning this lab such as analysis design, logical, movement of machine, programming skill and community. The basically I just know about machine mechanism when speed of each motor. For example,

1. If Speed Motor Left = Right && >= 1 : The robot run "Straight Forward"
2. If Speed Motor Left = Right && <= -1 : The robot run "Straight Backward"



```
def move(self, velocity):
    # velocity < 0 = reverse
    # velocity > 0 = forward
    res = sim.simxSetJointTargetVelocity(clientID, self.leftMotor,
    velocity, sim.simx_opmode_blocking)
    res = sim.simxSetJointTargetVelocity(clientID, self.rightMotor,
    velocity, sim.simx_opmode_blocking)
```

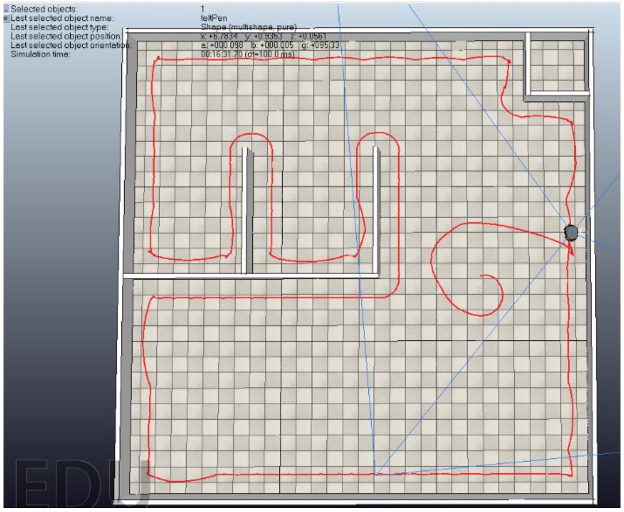
3. If Speed Motor Left > Right : The robot turn "Right"
4. If Speed Motor Left < Right : The robot turn "Left"



```
def turn(self, turnVelocity):
    # turnVelocity < 0 = trun left
    # turnVelocity > 0 = turn right
    res = sim.simxSetJointTargetVelocity(clientID, self.leftMotor,
    turnVelocity, sim.simx_opmode_blocking)
    res = sim.simxSetJointTargetVelocity(clientID, self.rightMotor,
    turnVelocity, sim.simx_opmode_blocking)
```

Part of coding, I do not have experience with Python language if I have a problem with coding. I solve a problem by searching on internet first after than if I do not get an answer, I will ask with classmate to describe a coding. However, I am a beginner of Python but I has been increased programming skill when I write a code. This project makes me know about step of programming and I really know some objects to must use and important. For example

1. Import library
2. Setup Motor (moving) and Sonars (Detecting)
3. Start Sonars and Sensor
4. Check API and Connecting
5. Setting Detected & Action after detected



Selected objects: 1
Last selected object name: feltPen
Last selected object type: Shape (multishape, pure)
Last selected object position: x: +6.7834 y: +0.9353 z: +0.0561
Last selected object orientation: a: +000.098 b: +000.005 g: +095.33
Simulation time: 00:16:31.70 (dt=100.0 ms)

The result and conclusions

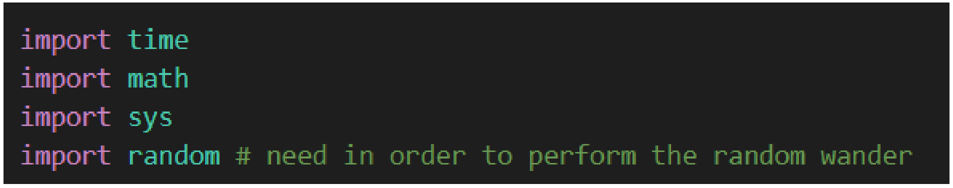
In the project start a robot do not have exit that it must run random position and when the robot detected wall. It must follow the wall that I have to understand three things before do the project. Firstly, machine mechanism such as how to make a robot run and how to make a robot turn. Secondly, design and analysis such a strategy when the robot run detected the wall what the robot should do. The lastly, programming skill (Python), I have to know how to write a code, syntax, when error how to solve, testing a code and logic when write code. However, I have to more practice and understand than before. Sometimes some issues are very difficult that I think but if I ask with each other, it would be very easy to solve. So, if I have a problem, I will search on internet first and may ask with each other or classmate to help.

What could have been done better, differently and in the future

On project I have many problems to solve because my experience is not enough to make it clear. I think that the robot should have a function to make a robot curve corner that it is not test by programing to set Left motor -2 and Right motor 2 to make turn such as firstly, a function curve corner 90 degree or 30 degree, if it has this function, it will save a time to coding because on process of turning. It spends time a lot (Testing). Lastly, the robot should have a battery life for practice a programming to design a movement of robot to develop for save energy like a moveless but has the most efficient.

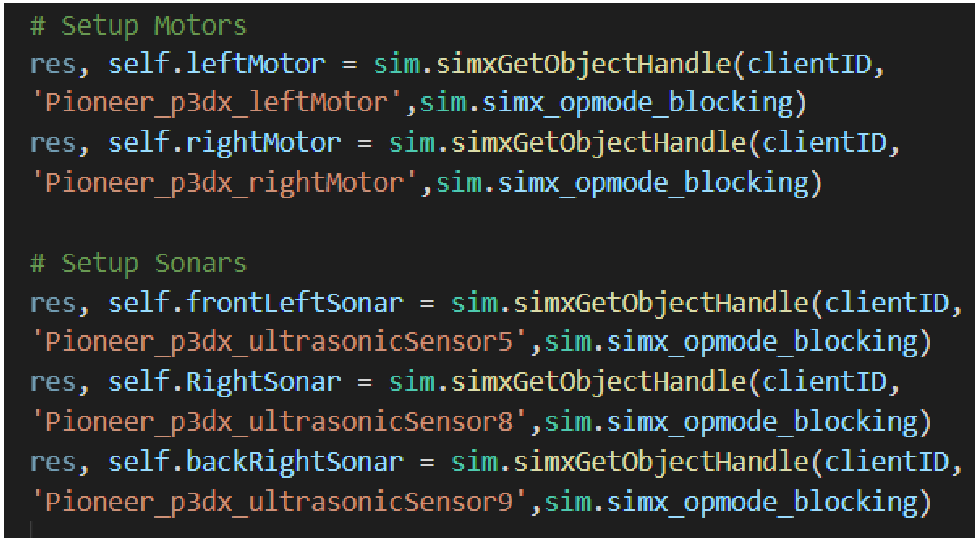
Appendix

1. Import library



```
import time
import math
import sys
import random # need in order to perform the random wander
```

2. Setup Motor (moving) and Sonars (Detecting)

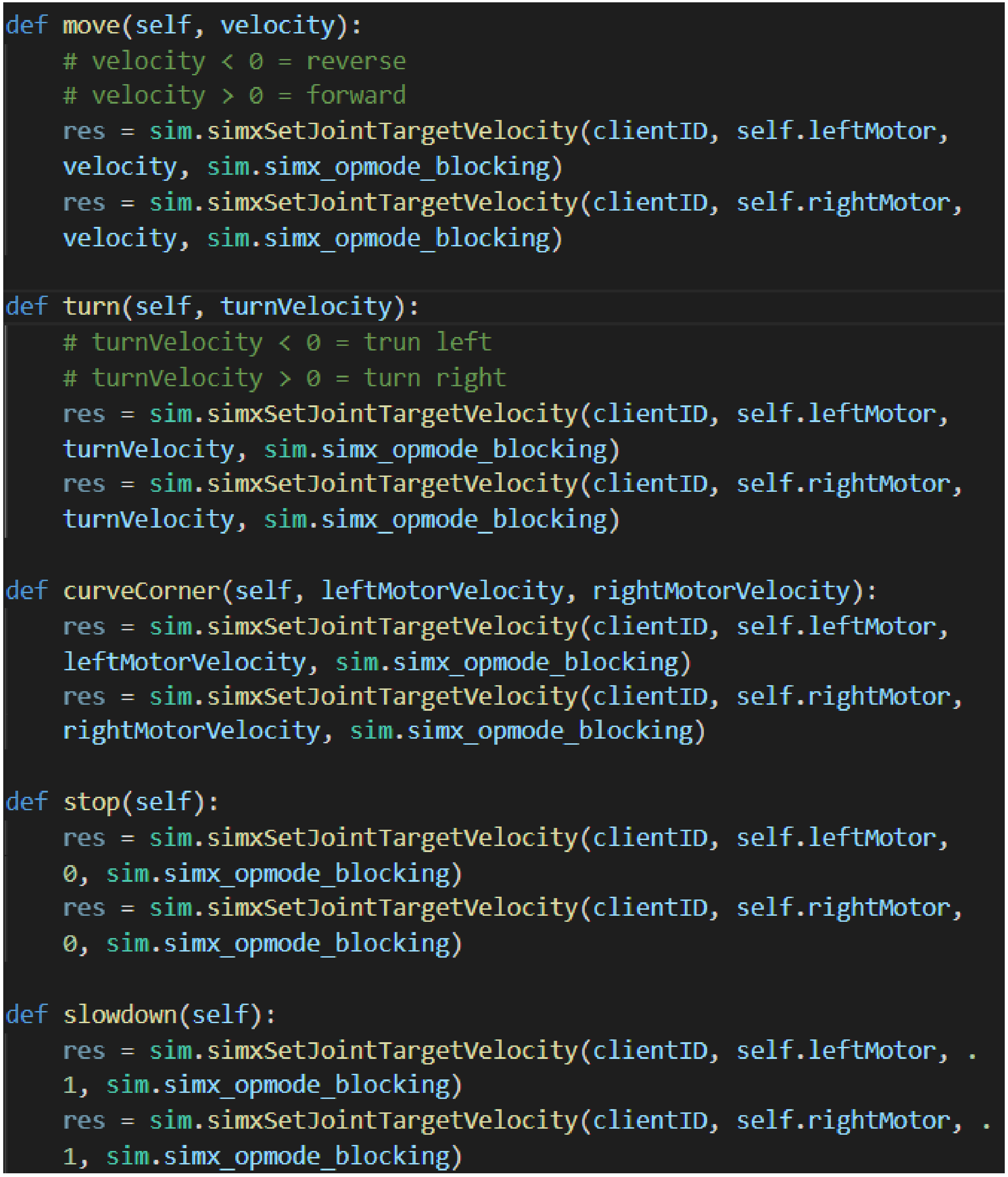


```
# Setup Motors
res, self.leftMotor = sim.simxGetObjectHandle(clientID,
'Pioneer_p3dx_leftMotor',sim.simx_opmode_blocking)
res, self.rightMotor = sim.simxGetObjectHandle(clientID,
'Pioneer_p3dx_rightMotor',sim.simx_opmode_blocking)

# Setup Sonars
res, self.frontLeftSonar = sim.simxGetObjectHandle(clientID,
'Pioneer_p3dx_ultrasonicSensor5',sim.simx_opmode_blocking)
res, self.RightSonar = sim.simxGetObjectHandle(clientID,
'Pioneer_p3dx_ultrasonicSensor8',sim.simx_opmode_blocking)
res, self.backRightSonar = sim.simxGetObjectHandle(clientID,
'Pioneer_p3dx_ultrasonicSensor9',sim.simx_opmode_blocking)
```

3. Start Sonars and Sensor

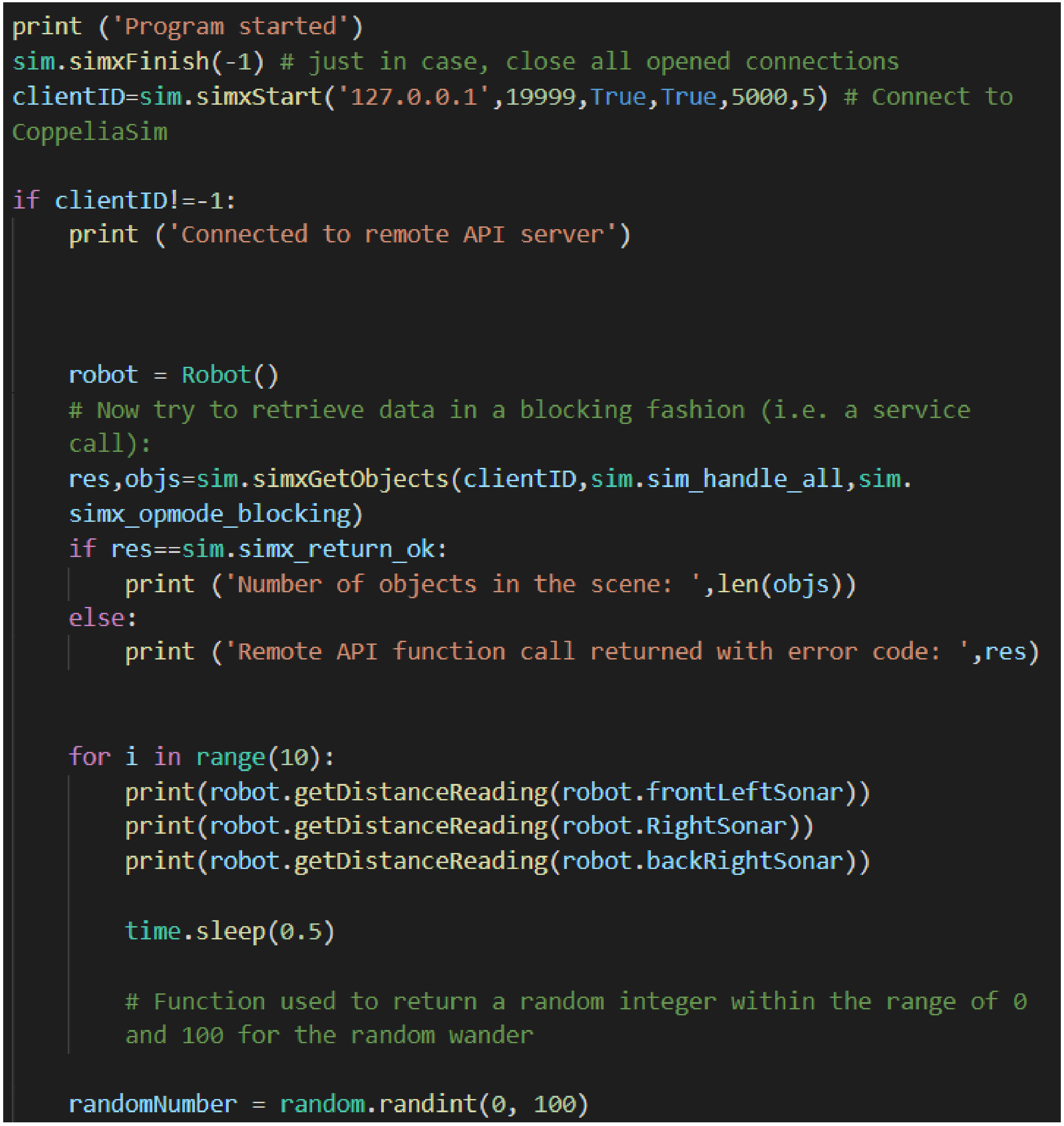
```
    # Start Sonars
    res,detectionState,detectedPoint,detectedObjectHandle,
    detectedSurfaceNormalVector = sim.simxReadProximitySensor
    (clientID,self.frontLeftSonar,sim.simx_opmode_streaming)
    res,detectionState,detectedPoint,detectedObjectHandle,
    detectedSurfaceNormalVector = sim.simxReadProximitySensor
    (clientID,self.RightSonar,sim.simx_opmode_streaming)
    res,detectionState,detectedPoint,detectedObjectHandle,
    detectedSurfaceNormalVector = sim.simxReadProximitySensor
    (clientID,self.backRightSonar,sim.simx_opmode_streaming)

    #Starting Sensors, front and back

def getDistanceReading(self, objectHandle):
    # Get reading from sensor
    res,detectionState,detectedPoint,detectedObjectHandle,
    detectedSurfaceNormalVector = sim.simxReadProximitySensor
    (clientID,objectHandle,sim.simx_opmode_buffer)
```

4. Create function (move, turn, curveCorner, stop)



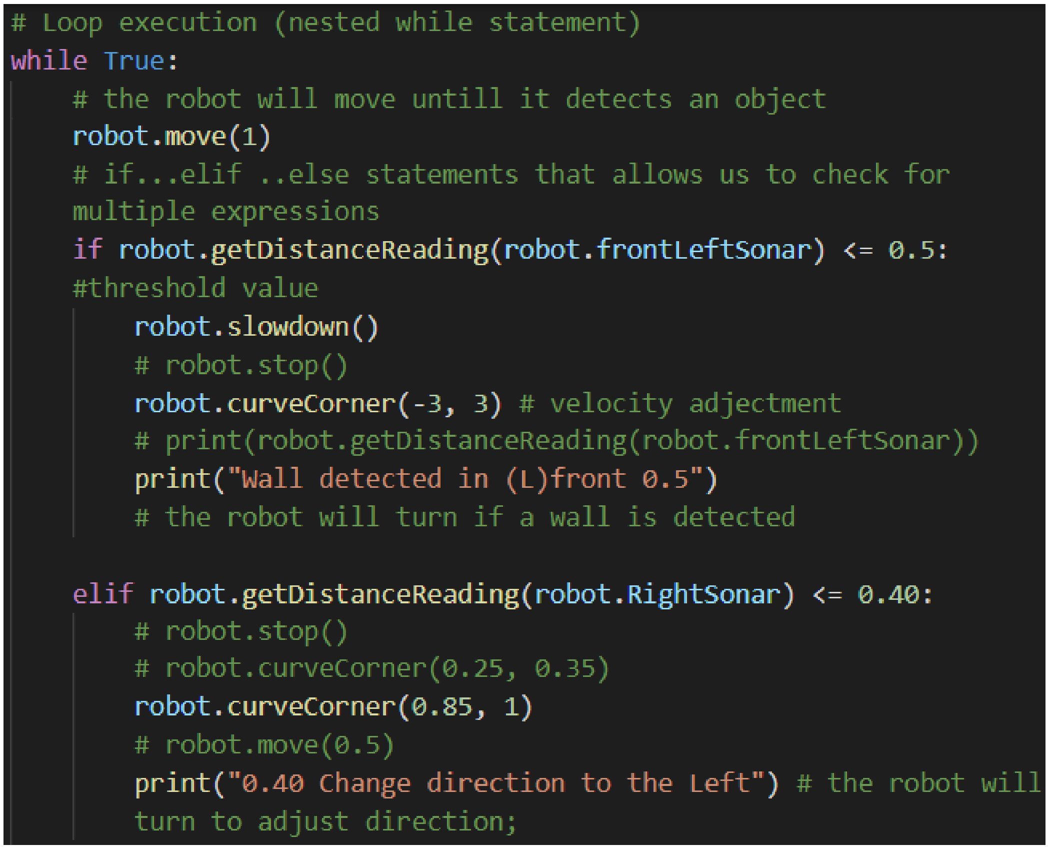
```
def move(self, velocity):
    # velocity < 0 = reverse
    # velocity > 0 = forward
    res = sim.simxSetJointTargetVelocity(clientID, self.leftMotor,
    velocity, sim.simx_opmode_blocking)
    res = sim.simxSetJointTargetVelocity(clientID, self.rightMotor,
    velocity, sim.simx_opmode_blocking)

def turn(self, turnVelocity):
    # turnVelocity < 0 = trun left
    # turnVelocity > 0 = turn right
    res = sim.simxSetJointTargetVelocity(clientID, self.leftMotor,
    turnVelocity, sim.simx_opmode_blocking)
    res = sim.simxSetJointTargetVelocity(clientID, self.rightMotor,
    turnVelocity, sim.simx_opmode_blocking)

def curveCorner(self, leftMotorVelocity, rightMotorVelocity):
    res = sim.simxSetJointTargetVelocity(clientID, self.leftMotor,
    leftMotorVelocity, sim.simx_opmode_blocking)
    res = sim.simxSetJointTargetVelocity(clientID, self.rightMotor,
    rightMotorVelocity, sim.simx_opmode_blocking)

def stop(self):
    res = sim.simxSetJointTargetVelocity(clientID, self.leftMotor,
    0, sim.simx_opmode_blocking)
    res = sim.simxSetJointTargetVelocity(clientID, self.rightMotor,
    0, sim.simx_opmode_blocking)

def slowdown(self):
    res = sim.simxSetJointTargetVelocity(clientID, self.leftMotor, .
    1, sim.simx_opmode_blocking)
    res = sim.simxSetJointTargetVelocity(clientID, self.rightMotor, .
    1, sim.simx_opmode_blocking)
```

5. Check API and Connecting

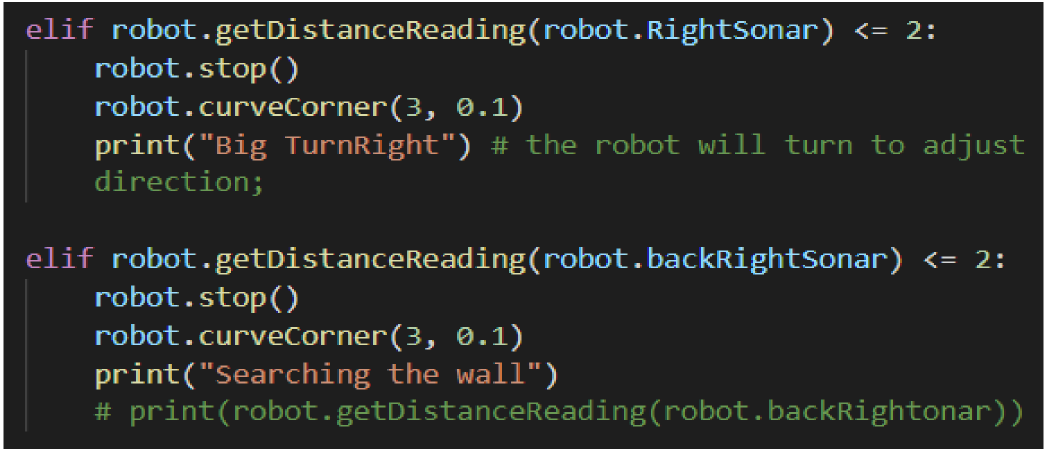


```
print ('Program started')
sim.simxFinish(-1) # just in case, close all opened connections
clientID=sim.simxStart('127.0.0.1',19999,True,True,5000,5) # Connect to
CoppeliaSim

if clientID!=-1:
    print ('Connected to remote API server')



    robot = Robot()
    # Now try to retrieve data in a blocking fashion (i.e. a service
    call):
    res,objs=sim.simxGetObjects(clientID,sim.sim_handle_all,sim.
    simx_opmode_blocking)
    if res==sim.simx_return_ok:
        print ('Number of objects in the scene: ',len(objs))
    else:
        print ('Remote API function call returned with error code: ',res)


    for i in range(10):
        print(robot.getDistanceReading(robot.frontLeftSonar))
        print(robot.getDistanceReading(robot.RightSonar))
        print(robot.getDistanceReading(robot.backRightSonar))

        time.sleep(0.5)

        # Function used to return a random integer within the range of 0
        and 100 for the random wander

    randomNumber = random.randint(0, 100)
```

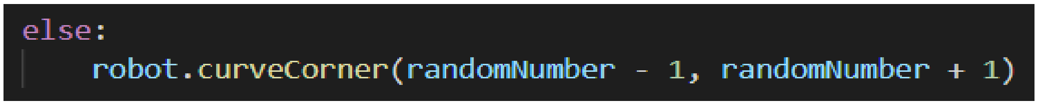
6. Setting Detected & Action after detected



```
# Loop execution (nested while statement)
while True:
    # the robot will move untill it detects an object
    robot.move(1)
    # if...elif ..else statements that allows us to check for
    multiple expressions
    if robot.getDistanceReading(robot.frontLeftSonar) <= 0.5:
    #threshold value
        robot.slowdown()
        # robot.stop()
        robot.curveCorner(-3, 3) # velocity adjectment
        # print(robot.getDistanceReading(robot.frontLeftSonar))
        print("Wall detected in (L)front 0.5")
        # the robot will turn if a wall is detected

    elif robot.getDistanceReading(robot.RightSonar) <= 0.40:
        # robot.stop()
        # robot.curveCorner(0.25, 0.35)
        robot.curveCorner(0.85, 1)
        # robot.move(0.5)
        print("0.40 Change direction to the Left") # the robot will
        turn to adjust direction;
```



```
elif robot.getDistanceReading(robot.RightSonar) <= 2:
    robot.stop()
    robot.curveCorner(3, 0.1)
    print("Big TurnRight") # the robot will turn to adjust
    direction;

elif robot.getDistanceReading(robot.backRightSonar) <= 2:
    robot.stop()
    robot.curveCorner(3, 0.1)
    print("Searching the wall")
    # print(robot.getDistanceReading(robot.backRightonar))
```



```
else:
    robot.curveCorner(randomNumber - 1, randomNumber + 1)
```