**Pi2Go Simulator Programming: WS24 Sample Answers**

**Exercise 1:** As with the previous worksheet, the lower epsilon\_reduce and learning\_rate are, the more likely the program is to learn the correct algorithm but the longer it will take. These values seem to work most of the time and its relatively quick.

import simclient.simrobot as pi2go

import time, random

pi2go.init()

actions = ['forward','backward','left','right']

def action\_reward(action\_list, default):

action\_rewards = {}

for i in range(0, 2):

for j in range(0, 2):

for k in (action\_list):

action\_rewards[((i, j), k)] = default

return action\_rewards

def execute\_action(action):

if (action == "forward"):

pi2go.forward(10)

elif (action == "backward"):

pi2go.reverse(10)

elif (action == "left"):

pi2go.spinLeft(10)

elif (action == "right"):

pi2go.spinRight(10)

else:

pi2go.stop()

time.sleep(3)

def best\_action(state):

max\_reward = 0

for act in actions:

if (reward\_dictionary[(state, act)] > max\_reward):

action = act

max\_reward = reward\_dictionary[(state, act)]

return action

reward\_dictionary = action\_reward(actions, 1)

rewards = {(1, 1):1, (1, 0):2, (0, 1):0, (0, 0):1}

epsilon = 1

epsilon\_reduce = 0.05

learning\_rate = 0.5

while (epsilon > 0):

explore = random.random()

state = (pi2go.irLeftLine(), pi2go.irRightLine())

if (explore < epsilon):

action = random.choice(actions)

print("Random Action: " + action)

else:

action = best\_action(state)

print("Best Action: " + action)

execute\_action(action)

reward = rewards[(pi2go.irLeftLine(), pi2go.irRightLine())]

reward\_dictionary[(state, action)] = reward\_dictionary[(state, action)] + (reward - reward\_dictionary[(state, action)])\*learning\_rate

if (reward == 2):

epsilon = epsilon - epsilon\_reduce

print("New epsilon: " + str(epsilon))

pi2go.stop()

print(reward\_dictionary)

for key in reward\_dictionary:

print("Average reward for " + str(key) + " is " + str(reward\_dictionary[key]))



University of Liverpool, 2020

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).