**Virtual Pi2Go Programming: Dictionaries and Tuples**

**AIM:** After completing this worksheet you should be able to use Python dictionary data structures in programs.

**You Need:** To complete this worksheet you need to have a virtual Initio simulator (see WS1), understand how to control the robot’s motors (WS3), be able to use files to store Programs (WS5), understand Python control structures (WS7-10), basic data types (WS12-15), lists (WS19) and the Time and Random module (WS6 & WS20)

**If the simulator isn’t already running: Start the Simulator, Select the Pi2go Simulation and square.xml, then start IDLE (open a *new IDLE window* if you have used IDLE to start the simulator).**

**Dictionaries** are a common data structure used in many programming languages. They let you *look up* a value based on some key word. So, for instance, you could associate each action the robot can take with the duration of the action:

action\_dictionary = {“forward”:3, “backward”:2, “left”:5, “right”:3, “stop”:2}

Each entry in the dictionary is a pair of a *key* and a *value* separated by a colon.

**Question 1:** In action\_dictionary what are the keys?

**Question 2:** What are the values?

You can access the value of an entry in a dictionary with the syntax *dictionary[key]*

Consider the following program.

import simclient.simrobot as pi2go

import time, random

pi2go.init()

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

action\_dictionary = {"forward":3, "backward":2, "left":5, "right":3, "stop":2}

while True:

action = random.choice(actions)

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(action\_dictionary[action])

**Question 3:** What does the program do?

You can loop through all the keys in a dictionary using

for *key* in *dictionary*:

This is similar to how you could loop through every element in a list.

**Exercise 1:** Modify the above program so that instead of selecting an action at random from the list actions it loops through each of the keys in action\_dictionary in turn and attempts that action for the set duration.

Write your program below.

You can change the value associated with a key in a dictionary with the syntax

*dictionary[key]* = *value*

Consider the following program:

import simclient.simrobot as pi2go

import time, random

pi2go.init()

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

dictionary = {"forward":0, "backward":0, "left":0, "right":0, "stop":0}

while True:

action = random.choice(actions)

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)

if (pi2go.irLeftLine()):

dictionary[action] = dictionary[action] + 1

if (pi2go.irRightLine()):

dictionary[action] = dictionary[action] + 1

if (dictionary[action] > 3):

break

pi2go.stop()

print(dictionary)

**Question 4:** What does the program do?

Trying running the program when the robot starts on the black square and when the robot starts some way from the black square.

**Question 5:** What difference does it make where the robot starts?

**Exercise 2:** Modify your program so that it tries 20 actions at random adding one to their *score* if either of the line sensors detects a line at the end of the action. After 20 actions the program prints out the action with the highest score.

**More on Keys:** So far, we have been using strings as the keys for our dictionaries, but it is also possible to use numbers as keys. You can’t use lists as keys, but you can use a special kind of list called a *tuple*. A tuple is like a list except that you can’t change the values in a tuple.

You can create a tuple with the syntax

*tuple* = (*element1, element2, element3*)

and access elements of a tuple as

*tuple*[*element*]

Consider the following program:

import simclient.simrobot as pi2go

import time, random

pi2go.init()

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

dictionary = {"forward":0, "backward":0, "left":0, "right":0, "stop":0}

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

while True:

action = random.choice(actions)

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)

dictionary[action] = dictionary[action] + rewards[(pi2go.irLeftLine(), pi2go.irRightLine())]

if (dictionary[action] > 3):

break

pi2go.stop()

print(dictionary)

**Question 6:** What does the rewards dictionary do?

**Exercise 3:** Modify the above program so that instead of the reward increasing when the line sensors detect a black surface, it increases when they detect a clear surface.



University of Liverpool, 2019

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