**Pi2Go Simulator Programming: File Input/Output**

**AIM:** After completing this worksheet you should be able to read information into a program from a file, and write data from a program to a file.

**You Need:** To complete this worksheet you need to have a virtual Pi2Go simulator (see WS1), understand how to control the robot’s motors and sensors (WS3&WS4), be able to use files to store Programs (WS5), use control structures (WS7-10), Data Types (WS12-14), functions (WS16), the time module (WS6) and have working programs from WS24.

**If the simulator isn’t already running: Start the Simulator, Select the Pi2Go Simulation and oval.xml, then start IDLE (open a *new IDLE window* if you have used IDLE to start the simulator).**

Once your machine learning program has learned which actions to take in some state it would be nice if it these could be saved and reused.

In order to do this, you need to be able to read and write data from files.

**Writing to Files**

In order to write to a file, you must first open the file, you can do this with the method:

>>> f = open(‘datafile’, ‘w’)

In the above call datafile is the name of the file and w is the *mode.* In this case w means the file is being opened in *write only mode*. **Open** returns a *file object* – so f is a file object for datafile.

Once a file is open for writing you can write to it using the method write(*string*)

Once you have finished writing to a file you should close it using f.close()

Consider the following program:

dictionary = {((1, 1), "forward"):0.5, ((1, 1), "backward"):0.5, ((1, 1), "left"):1, ((1, 1), "right"):0, ((1, 0), "forward"):1, ((1, 0), "backward"):0.5, ((1, 0), "left"):0, ((1, 0), "right"):0, ((0, 1), "forward"):0.5, ((0, 1), "backward"):0.5, ((0, 1), "left"):0, ((0, 1), "right"):1, ((0, 0), "forward"):0.5, ((0, 0), "backward"):0.5, ((0, 0), "left"):0, ((0, 0), "right"):1}

f = open(‘policy\_file’, 'w')

for key in dictionary:

f.write(str(key))

f.write(“::”)

f.write(str(dictionary[key]))

f.write('\n')

f.close()

**Note:** \n is the “new line” symbol, so when you print or write \n the next print or write command appears on a new line.

What does the program do?

**Exercise 1:** Adapt your machine learning program from WS24 so that it writes the policy it has learned to a file.

**More on File Modes:** There are four file modes:

* w – Write Only
* r – Read Only
* a – Append new data to the end of the file
* r+ -- Read and Write

**Reading from Files:** There are several ways to read information from a file, f:

* You can read all the information from at once using the function f.read()
* You can read the data one line at a time using the function f.readline()
* You can loop through the lines of the file using: for line in f – each time the program goes around the loop, the variable line will take the value of the next line in the file.

**Exercise 2:** Write a program that will read in the data from policy\_file that was created in the first program in this worksheet:

**Turning Strings in to Values:** If we are successfully going to use the information stored in our policy\_file then we are going to have extract the information about state, action and reward from our strings and turn them back into numbers and tuples. There are a number of functions that we can use for this.

* **Treating strings like lists:** We can treat strings like lists and use the [n] syntax to extract the *nth* character from the string for instance,

Try typing the following:

>>> s = “(0, 1)”

>>> s[0]

>>> s[1]

What happens and why?

* **Extracting ranges:** We can extract a range of letters from a string using the syntax: [a:b] where a is the first letter to be extracted and b is the last.

Try typing the following:

>>> s = “(0, 1)”

>>> s[1:4]

What happens and why?

* **Splitting strings.** Lastly, we can *split strings* by nominated a character and turning the string into an array of strings which appear between each appearance of that character.
* Sometimes when using split you need to use a backslash in front of the character if it is a quotation mark or similar that might indicate the end of the string.

Try typing the following:

>>> s = “(0, 1)”

>>> s.split(‘,’)

What happens and why?

Consider the following program:

f = open('policy\_file', 'r')

for line in f:

[key, value] = line.split('::')

[a, b, c] = key.split(',')

irR = a[2]

irL = b[1]

[x, action, y] = c.split('\'')

print(irR)

print(irL)

print(action)

print(value)

What does the program do?

**Exercise 3:** Write a program that will read in the policy from policy\_file and then use it to control the Pi2Go to follow the black oval until the switch is pressed.

**Hint:** You will need to use int and float to convert strings to integers and floats.



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/).