CS 2302 – Project 1, Option A
Professor Diego Aguirre
Victor Huicochea – 80643271

Project 1 – Recursion

Introduction

In a shared Google Drive folder, people uploaded several pictures of dogs and cats. The problem is that some of them uploaded folders inside this main folder, and some others uploaded more folders inside the second folder and so on. Pictures can be found inside any folder, not only in the very last one of the directory trees.

A program must be written such that it traverses through every possible folder using recursion and finds every possible image to sort them into two lists: dogs and cats.

Proposed Solution

The implemented method works using for loops in recursive calls. For every new folder/directory found in the current directory, it will keep calling the recursive method until it reaches a final folder. When this final folder is reached, the method creates two temporary lists, one for dog pictures and one for cat pictures. The method decides where to store each picture using the classify_pic method provided by the instructor.

After storing these two temporary lists, it will go one folder back, if there is another folder in the directory that the method just returned, it will go inside it until it reaches again a final folder. Then it will repeat the same operations: create to temporary lists that will be passed to the mother folder. It will be tracing back until it reaches the original folder, delivering a complete list, which basically is the addition of every temporary list.

Experimental Results

1.- Testing with the folders provided by the instructor.

The Dropbox zip folder was downloaded and extracted to be used as the first test of the program. The “Main” program was placed inside the directory labelled “1.1 CatsDogs”, in this same directory there was another directory labelled as “Pictures”.

Output after running program:

"D:\Victor\Documents\CS\CS 2302\venv\Scripts\python.exe" "D:/Victor/Documents/CS/CS 2302/Lab1/1.1 CatsDogs/1.1 CatsDogs/Main.py"

.//Pictures/DCIM/100ANDRO/cat.25.jpg

.//Pictures/DCIM/100ANDRO/cat.26.jpg

.//Pictures/DCIM/cat.19.jpg

.//Pictures/DCIM/cat.20.jpg

.//Pictures/Maria/Pics/CatsAndDogs/cat.82.jpg

.//Pictures/Maria/Pics/CatsAndDogs/cat.83.jpg

.//Pictures/Maria/Pics/CatsAndDogs/cat.84.jpg

.//Pictures/Maria/Pics/CatsAndDogs/cat.85.jpg

.//Pictures/Maria/Pics/CatsAndDogs/cat.86.jpg

.//Pictures/Maria/Pics/cat.58.jpg

.//Pictures/Maria/Pics/cat.59.jpg

.//Pictures/Maria/cat.31.jpg

.//Pictures/Maria/cat.32.jpg

.//Pictures/Maria/cat.33.jpg

.//Pictures/Maria/cat.34.jpg

.//Pictures/Pics/All/cat.128.jpg

.//Pictures/Pics/All/cat.129.jpg

.//Pictures/Pics/All/cat.130.jpg

.//Pictures/Pics/cat.102.jpg

.//Pictures/Pics/cat.103.jpg

.//Pictures/Pics/cat.104.jpg

.//Pictures/Pics/cat.105.jpg

.//Pictures/cat.11.jpg

.//Pictures/cat.12.jpg

.//Pictures/cat.13.jpg

.//Pictures/cat.14.jpg

.//Main.py

.//Pictures/DCIM/100ANDRO/dog.101.jpg

.//Pictures/DCIM/100ANDRO/dog.102.jpg

.//Pictures/DCIM/100ANDRO/dog.103.jpg

.//Pictures/DCIM/100ANDRO/dog.104.jpg

.//Pictures/DCIM/100ANDRO/dog.105.jpg

.//Pictures/DCIM/dog.20.jpg

.//Pictures/DCIM/dog.6.jpg

.//Pictures/DCIM/dog.7.jpg

.//Pictures/DCIM/dog.8.jpg

.//Pictures/DCIM/dog.9.jpg

.//Pictures/Maria/Pics/CatsAndDogs/dog.304.jpg

.//Pictures/Maria/Pics/CatsAndDogs/dog.305.jpg

.//Pictures/Maria/Pics/dog.274.jpg

.//Pictures/Maria/Pics/dog.275.jpg

.//Pictures/Maria/Pics/dog.276.jpg

.//Pictures/Maria/Pics/dog.277.jpg

.//Pictures/Maria/Pics/dog.278.jpg

.//Pictures/Maria/Pics/dog.279.jpg

.//Pictures/Maria/dog.159.jpg

.//Pictures/Maria/dog.160.jpg

.//Pictures/Maria/dog.161.jpg

.//Pictures/Maria/dog.162.jpg

.//Pictures/Maria/dog.163.jpg

.//Pictures/Maria/dog.164.jpg

.//Pictures/Maria/dog.165.jpg

.//Pictures/Pics/All/dog.344.jpg

.//Pictures/Pics/All/dog.345.jpg

.//Pictures/Pics/All/dog.346.jpg

.//Pictures/Pics/All/dog.347.jpg

.//Pictures/Pics/dog.308.jpg

.//Pictures/Pics/dog.309.jpg

.//Pictures/Pics/dog.310.jpg

.//Pictures/dog.367.jpg

.//Pictures/dog.368.jpg

Process finished with exit code 0

2.- Testing with empty folder

The program was placed inside an empty folder to verify that it would provide a useful message for when no pictures were found.

Output after running program:

"D:\Victor\Documents\CS\CS 2302\venv\Scripts\python.exe" "D:/Victor/Documents/CS/CS 2302/Lab1/1.1 CatsDogs/Test2/Main.py"

There are no available cats to adopt!

There are no available dogs to adopt!

Process finished with exit code 0

3.- Testing when there are only cat pictures

The program was placed inside a directory tree which only contained cat pictures to ensure that it would provide a useful message when no dogs pictures were found.

Output after running program:

"D:\Victor\Documents\CS\CS 2302\venv\Scripts\python.exe" "D:/Victor/Documents/CS/CS 2302/Lab1/1.1 CatsDogs/Test2/Main.py"

.//Pics/All/cat.128.jpg

.//Pics/All/cat.129.jpg

.//Pics/All/cat.130.jpg

.//Pics/cat.102.jpg

.//Pics/cat.103.jpg

.//Pics/cat.104.jpg

.//Pics/cat.105.jpg

.//Main.py

There are no available dogs to adopt!

Process finished with exit code 0

4.- Testing when there are only dog pictures

The program was placed inside a directory tree which only contained dog pictures to ensure that it would provide a useful message when no dogs pictures were found.

Output after running program:

"D:\Victor\Documents\CS\CS 2302\venv\Scripts\python.exe" "D:/Victor/Documents/CS/CS 2302/Lab1/1.1 CatsDogs/Test2/Main.py"

There are no available cats to adopt!

.//Maria/Pics/CatsAndDogs/dog.304.jpg

.//Maria/Pics/CatsAndDogs/dog.305.jpg

.//Maria/Pics/dog.274.jpg

.//Maria/Pics/dog.275.jpg

.//Maria/Pics/dog.276.jpg

.//Maria/Pics/dog.277.jpg

.//Maria/Pics/dog.278.jpg

.//Maria/Pics/dog.279.jpg

.//Maria/dog.159.jpg

.//Maria/dog.160.jpg

.//Maria/dog.161.jpg

.//Maria/dog.162.jpg

.//Maria/dog.163.jpg

.//Maria/dog.164.jpg

.//Maria/dog.165.jpg

Process finished with exit code 0

Conclusions

This project helped me to completely understand how does recursion work. I'm now not only able to recognize what it does, but I can now also visualize how the program moves and how data is passed from every iteration to the next one.

A second key knowledge obtained from this project is the Python syntax. Since this is the first class I have taken in Python, I was not comfortable using it at the beginning. After finishing this assignment, I can now say that I feel much more confident to write Python programs.

Appendix – Source Code

```
# Course: CS2301 - Data Structures
# Author: Victor Huicochea
# Assignment: Project 1 - Option A
# Instructor: Diego Aguirre
# T.A.: Manoj Saha
# Last date of modification: 09/12/2018
# Purpose: Practice recursion in a real-life situation.


import os
import random


# Method provided by instructor. It returns two lists, one
holding the names of all the folders in the current folder
#                                 and the other one holding the
names of all files inside the current folder.
def get_dirs_and_files(path):
    dir_list = [directory for directory in os.listdir(path) if os
.path.isdir(path + '/' + directory)]
    file_list = [directory for directory in os.listdir(path) if
not os.path.isdir(path + '/' + directory)]

    return dir_list, file_list


# Method provided by instructor. It returns a number between 0
and 1. The closer it is to 1, the more likely the file
#                                 located in the given path is a
dog picture. The closer it is to 0, the more likely it
#                                 is a cat picture.
def classify_pic(path):
    # To be implemented by Diego: Replace with ML model
    if "dog" in path:
        return 0.5 + random.random() / 2

    return random.random() / 2


# Method defined by student. Method uses recursion to explore
until the last directory and then starts saving the paths
#                             to each file in its respective list.
def process_dir(path):
    dir_list, file_list = get_dirs_and_files(path)

    cat_list = []
```

```
    dog_list = []

    # Your code goes here
    # Checks if there is any other directory in current directory
.
    if len(dir_list) > 0:
        for i in range(len(dir_list)):
            new_cats, new_dogs = process_dir(path + '/' + dir_
list[i])  # recursive call with updated path
            for j in range(len(new_cats)):  # Assignment of new
list from current directory to the master list
                cat_list.append(new_cats[j])
            for k in range(len(new_dogs)):  # Assignment of new
list from current directory to the master list
                dog_list.append(new_dogs[k])

    for i in range(len(file_list)):
        if classify_pic(path + '/' + file_list[i]) >= .5:  #
Creation of new list in current directory
            dog_list.append(path + '/' + file_list[i])
        else:
            cat_list.append(path + '/' + file_list[i])

    return cat_list, dog_list


def main():
    start_path = './'  # current directory

    cats, dogs = process_dir(start_path)

# Loops created to print values inside each list
    for m in range(len(cats)):
        print(cats[m])
    for n in range(len(dogs)):
        print(dogs[n])


main()
```

Academic Honesty Certification

I certify that this project is entirely my own work. I wrote, debugged, and tested the code being presented, performed the experiments, and wrote the report. I also certify that I did not share my code or report or provided inappropriate assistance to any student in the class.