Kyle Biondich

8/20/2023

IT FDN 110 A

Assignment 07

https://kbiondo.github.io/IntroToProg-Python-Mod07/

Pickling with Errors

Introduction

Week 7 of the course introduced pickling and error handling in python. The following paragraphs outline the examples I found on the internet and the methods that were used to create a game save state into a state.bin file as well as how the try, exception, and finally error handling block statements work.

Intended Outcome

The intended outcome of this week is the example of pickling game save states and error handling.

```
PS D:\DEV\uw\assignments\_PythonClass\Assignment07> python .\Assignment07_KyleBiondich.py
End of file error
Player coordinates: (3, 2)
Obstacles: {(7, 4), (1, 1), (0, -1), (5, 6)}
Number of items: 2
remember, this list is 3 elements long
Second element = 2
uh oh, you chose a 4th element
no no divide by zero = bad
Finally is always executed even if you try to divide by zero, which is bad
PS D:\DEV\uw\assignments\_PythonClass\Assignment07>
```

Figure 1: Intended Outcome: Game Information and Error Handling

Learning to Pickle

After a quick googling, I stumbled upon this website (The ultimate guide to Python pickle, n.d.). I used its example of how to pickle a game save state to a saveState.bin file, and then read that file back into the script. I chose this site because it appeared to have a short and easy example that I could use. I unfortunately ran into some issues with it and ended up having to use some of the try, except handling

from the second part of the script. Technically my script runs fine, but it could use some work with the 'EndOFFIle' issue it seems to be creating. I'll have to continue researching.

Initially, this example creates a class of GameItem and a second class of GameState. It gives a player, an obstacle, and items an initial state and then passes those to a state variable. From there, a saveState variable is created to hold the open function and an argument of 'wb' write binary is passed to it. Then pickle.dump reads the state object into a 'saveState' binary file.

Next, saveState is passed the open function with an argument of 'rb' read binary and the saveState.bin file is opened and passed into savedState. From here I get an EOFError and use the try and except handling to either open and read the file contents or to use the previous values of state.

I need to do a little more research on why the EOF error is occurring.

```
17  import pickle
18
19  # To pickle a file into an object:
20
21  # This example from snyk creates an example game that dumps the state of the game into a file called state.bin
    You, 2 days ago | 1 author (You)
22  class GameItem:
23      def __init__(self, name, cost):
24          self.name = name
25          self.cost = cost
26
27  # this is an example of how a game might save a play state
    You, 2 days ago | 1 author (You)
28  class GameState:
29      def __init__(self, player_coordinates, obstacles, items):
30          self.player_coordinates = player_coordinates
31          self.obstacles = obstacles
32          self.items = items
33
34  # Constructing a specifc state and pickling it into an object:
35  player_coordinates = (3, 2)
36  obstacles = { (1, 1), (5, 6), (7, 4), (0, -1) }
37  items = [ GameItem("Sword", 500), GameItem("Potion", 150) ]
38
39   # "wb" because we want to write in binary mode
40  state = GameState(player_coordinates, obstacles, items)
41  saveState = open("saveState.bin", "wb")
42  pickle.dump(state, saveState)
43
44  # To unpickle a file into an object:
45  # "rb" because we want to read in binary mode
46  savedState = open("saveState.bin", "rb")
47  try:
48      state = pickle.load(savedState)
49  except EOFError:
50      print("End of file error")
51      state = GameState(player_coordinates, obstacles, items)
52
53  print("Player coordinates:", state.player_coordinates)
54  print("Obstacles:", state.obstacles)
55  print("Number of items:", len(state.items))
56
```

Figure 2: Learning to Pickle

Learning to Handle Errors

I then googled a few sites about handling errors and came across this website (Python Exception Handling, n.d.). The example creates a three element list of numbers and then starts a try block and attempts to find an element at position 4. The except portion of the try block catches the error and explains that issue.

Another example is the divide by zero error. This try block presents x = 10/0 and catches the error with the 'ZeroDivisionError' exception. This example also uses a 'Finally' block which executes regardless of the exceptions above.

```
57  # ------------------------------------------------------------------------ #
58  # Learning how to do error handling
59  # From: https://wwww.geeksforgeeks.org/python-exception-handling/
60  # this site explains some of the types of error handling that is built into python and how to use them effectively.
61
62  # Simple error to handle runtime issues
63  a = [1, 2, 3]
64  try:
65      print('remember, this list is %d elements long' %(len(a)))
66      print ("Second element = %d" %(a[1]))
67          You, 23 hours ago • hopefully completed this 07 hw …
68      # Throws error since there are only 3 elements in array. Position 3 is actually the 4th element.
69      print ("Fourth element = %d" %(a[3]))
70
71  except:
72      print ("uh oh, you chose a 4th element")
73
74
75  # another error example but with using of the finally exception clause:
76
77  # No exception Exception raised in try block
78  try:
79      x = 10/0
80      print(x)
81
82  #this handles the divide by zero exception
83  except ZeroDivisionError:
84      print("no no divide by zero = bad")
85
86  finally:
87      # this block is always executed
88      # regardless of exception generation.
89      print('Finally is always executed even if you try to divide by zero, which is bad')
```

Figure 3: Handling Errors

Observations

This week's objective was interesting. Pickling can be a useful resource for saving obscured data which I may use at work. Although I haven't quite figured out how to solve the End OF File error, I see this as a useful exercise. The error handling portion of the objective is also useful to understand, especially as I build applications for work. Another big understanding is the vast amount of examples around the web. I found that I could google and find examples for everything I was trying to accomplish.

Summary

In summary, utilizing all the resources provided to the class and the online lecture, this paper outlines all the steps that were taken to create a python script that results in a successful execution of the intended outcome (Figure 1). Following the steps outlined above will allow for the audience to recreate the presented result.

References

Dictionaries. (n.d.). Retrieved from AfterHours Programming: https://www.afterhoursprogramming.com/tutorial/

Python Exception Handling. (n.d.). Retrieved from Geeks for Geeks: https://www.geeksforgeeks.org/python-exception-handling/

The ultimate guide to Python pickle. (n.d.). Retrieved from snyk.io: https://snyk.io/blog/guide-to-python-pickle/