import glob

import csv

from random import choice

import interface

import explorer

def get\_new\_csv\_data\_dict(new\_data\_file=None, ask\_user=True, extra\_info=True, \*\*kwargs):

"""

Return a new `csv\_data\_dict` based on a default dict.

The `new\_csv\_dict` function variable should contain keys and default values

expected by any functions called by the main menu interface. Add new entries

to it when expanded program functionality will expect all CSV dictionaries to

contain them by default.

Optional argument extra\_info controls whether summary information is printed.

Optional keyword argument ask\_user is passed to `get\_selected\_columns\_list`.

"""

if new\_data\_file == None:

filename = interface.choose\_csv()

else:

filename = new\_data\_file

# Dictionary with all keys expected by default functions

row\_dict\_list = get\_row\_dict\_list(filename, \*\*kwargs)

kwargs["row\_dict\_list"] = row\_dict\_list

sample\_row = choice(row\_dict\_list)

columns = sample\_row.keys()

new\_csv\_dict = {

"filename" : filename,

"row\_dict\_list" : row\_dict\_list,

"sample\_row" : sample\_row,

"number\_of\_rows" : len(row\_dict\_list),

"number\_of\_columns" : len(columns),

"columns" : columns,

"selected\_columns" : get\_selected\_columns\_list(ask\_user=ask\_user,\*\*kwargs),

"column\_type\_dict" : dict(zip(columns,[str for x in columns]))

}

if extra\_info:

explorer.print\_csv\_info\_from\_dict(new\_csv\_dict)

return new\_csv\_dict

def get\_row\_dict\_list(filename\_str='ischools-clean.csv', \*\*kwargs):

"""

Returns a list of dictionaries from a CSV file.

Row dictionaries take keys from the CSV column header. Values are the data in that column of the row.

Optional `filename\_str` parameter changes the file read in.

"""

with open(filename\_str, newline='') as file:

reader = csv.DictReader(file)

row\_dict\_list = list(reader)

return row\_dict\_list

def get\_csv\_filenames\_list():

"""

Returns a list of .csv files visible to the program.

"""

return glob.glob(r"\*.csv")

def get\_selected\_columns\_list(row\_dict\_list, ask\_user = True, \*\*kwargs):

"""

Get a list of columns to include when printing CSV data.

Columns are keys of the first dictionary in row\_list

If `ask\_user` is `True`, prompt the user for each column heading.

Otherwise, inlcude all columns.

Returns a list of column name strings.

"""

first\_row = row\_dict\_list[0]

columns = list(first\_row.keys())

print(f"There are {len(columns)} columns.")

if ask\_user and interface.input\_is\_yes(f"Select which to print?", default='y'):

selected\_columns = []

for column in columns:

selected =interface.input\_is\_yes(f" Do you want {column} in simplified print view?", default = 'n')

if selected:

selected\_columns.append(column)

else:

print(f"All columns will be printed.")

selected\_columns = columns

return selected\_columns

def get\_selected\_columns\_list\_in\_dict(\*\*kwargs):

"""

Calls `get\_selected\_columns\_list` and returns result as dictionary compatible

with main menu architecture.

"""

return {

"selected\_columns": get\_selected\_columns\_list(\*\*kwargs)

}