

```

1  """table.py | Robin Forestier | 15.03.2022
2
3  This file contains the two class for create the two table in the settings page.
4  """
5
6  # Importing the module `pickle` which allows saving and load python objects.
7  import pickle
8
9
10 # The Table class is a container for the data in a table
11 class Table:
12     def __init__(self):
13         """The function is used to initialize the class"""
14
15         # The function is used to load the new_colonne_heure variable from the
16         # pickle file
17         try:
18             self.colonne_heure = self.get_colonne_heure()
19             self.new_colonne_heure = self.get_new_colonne_heure()
20             self.table = self.get_table()
21         except FileNotFoundError:
22             self.colonne_heure = []
23             self.new_colonne_heure = []
24             self.table = []
25
26     def create(self):
27         """This function is used to create the table if it doesn't exist"""
28         self.colonne_heure = [{"{:02d}:00".format(i)} for i in range(24)]
29         self.table = [[0 for i in range(7)] for j in range(24)]
30
31     def modify_table(self, row, column):
32         """If the value of the cell is 3, change it to 0. Otherwise, add 1 to the
33         value of the cell
34
35         :param row: The row of the cell you want to change
36         :type row: int
37         :param column: The column of the cell you want to change
38         :type column: int
39         :return: The modified table.
40         :rtype: list
41         """
42
43         if self.table[row][column] == 3:
44             self.table[row][column] = 0
45         else:
46             self.table[row][column] += 1
47
48         return self.table
49
50     def add_custom_line(self, heure):
51         """Add a new line to the table if it doesn't already exist
52
53         :param heure: the time you want to add
54         :type heure: str
55         :return: The table and the list of the hours.
56         :rtype: list
57         """
58
59         existing_heure = self.get_colonne_heure()
60
61         if heure not in existing_heure:
62             self.colonne_heure.append(heure)
63             self.colonne_heure.sort()
64
65             self.set_colonne_heure()
66
67             self.new_colonne_heure.append(heure)
68             self.new_colonne_heure.sort()
69
70             self.set_new_colonne_heure()
71
72             index = self.colonne_heure.index(heure)

```

```

71         self.table.insert(index, [0 for i in range(7)])
72
73         self.set_table()
74
75         return self.table, self.colonne_heure
76
77     def del_custom_line(self, heure):
78         """It deletes the line of the table corresponding to the given heure
79
80         :param heure: the hour you want to delete
81         :type heure: str
82         """
83         index = self.new_colonne_heure.index(heure)
84         self.new_colonne_heure.pop(index)
85
86         index = self.colonne_heure.index(heure)
87         self.colonne_heure.pop(index)
88
89         self.table.pop(index)
90
91         self.set_new_colonne_heure()
92         self.set_colonne_heure()
93         self.set_table()
94
95     def set_table(self):
96         """This function is used to save the table to a pickle file"""
97
98         with open('./table/settings/table.pickle', 'wb') as f:
99             pickle.dump(self.table, f, pickle.HIGHEST_PROTOCOL)
100
101     def get_table(self):
102         """It loads the table from the pickle file
103
104         :return: The table
105         :rtype: list
106         """
107
108         try:
109             with open('./table/settings/table.pickle', 'rb') as f:
110                 # The protocol version used is detected automatically, so we do not
111                 # have to specify it.
112                 self.table = pickle.load(f)
113
114             return self.table
115         except FileNotFoundError:
116             self.create()
117             self.set_table()
118             self.set_colonne_heure()
119
120             self.get_table()
121
122     def set_colonne_heure(self):
123         """The function is used to save the value of the variable "colonne_heure" in
124         a file called "heure.pickle" """
125
126         with open('./table/settings/heure.pickle', 'wb') as f:
127             pickle.dump(self.colonne_heure, f, pickle.HIGHEST_PROTOCOL)
128
129     def get_colonne_heure(self):
130         """The function is used to load the value of the variable "colonne_heure"
131         from a file called "heure.pickle
132
133         :return: The list of the hours
134         :rtype: list
135         """
136
137         try:
138             with open('./table/settings/heure.pickle', 'rb') as f:
139                 # The protocol version used is detected automatically, so we do not
140                 # have to specify it.
141                 self.colonne_heure = pickle.load(f)
142
143             return self.colonne_heure

```

```

141         except FileNotFoundError:
142             self.create()
143             self.set_table()
144             self.set_colonne_heure()
145
146             self.get_colonne_heure()
147
148     def set_new_colonne_heure(self):
149         """The function is used to save the new_colonne_heure variable in a pickle
150         file"""
151
152         with open('./table/settings/new_heure.pickle', 'wb') as f:
153             pickle.dump(self.new_colonne_heure, f, pickle.HIGHEST_PROTOCOL)
154
155     def get_new_colonne_heure(self):
156         """The function is used to get the new_colonne_heure from the pickle file
157
158         :return: The new_colonne_heure variable is being returned
159         :rtype: list
160         """
161
162         try:
163             with open('./table/settings/new_heure.pickle', 'rb') as f:
164                 # The protocol version used is detected automatically, so we do not
165                 # have to specify it.
166                 self.new_colonne_heure = pickle.load(f)
167
168             return self.new_colonne_heure
169         except FileNotFoundError:
170             self.set_new_colonne_heure()
171             self.get_new_colonne_heure()
172
173 # It inherits from Table and overrides the __init__ method.
174 class Table2(Table):
175     def __init__(self):
176         """The __init__ function is called when an instance of the class is created.
177         used to initialize the attributes of the class.
178         """
179
180         try:
181             self.table = self.get_table()
182             self.name = self.get_name()
183         except FileNotFoundError:
184             self.table = []
185             self.name = []
186
187     def create(self):
188         """Create a table with 8 rows and 7 columns and create the name column."""
189
190         self.table = [[0 for i in range(7)] for j in range(8)]
191         self.name = ["Ligne 1", "Ligne 2", "Ligne 3", "Ligne 4", "Ligne 5", "Ligne
192         6", "Ligne 7", "couloir"]
193
194     def modif_table(self, row, column):
195         """Invert the value of a cell in the table
196
197         :param row: The row of the cell you want to change
198         :type row: int
199         :param column: The column of the cell you want to change
200         :type column: int
201         :return: The modified table
202         :rtype: list
203         """
204
205         if self.table[row][column] == 1:
206             self.table[row][column] = 0
207         else:
208             self.table[row][column] = 1
209
210         return self.table

```

```

211 def set_table(self):
212     """The function opens the pickle file and dumps the table into the pickle
        file"""
213
214     with open('./table/settings/table2.pickle', 'wb') as f:
215         pickle.dump(self.table, f, pickle.HIGHEST_PROTOCOL)
216
217 def get_table(self):
218     """Load the table from the pickle file
219
220     :return: A table object
221     :rtype: list
222     """
223
224     try:
225         with open('./table/settings/table2.pickle', 'rb') as f:
226             # The protocol version used is detected automatically, so we do not
227             # have to specify it.
228             self.table = pickle.load(f)
229
230             return self.table
231     except FileNotFoundError:
232         self.create()
233         self.set_table()
234         self.set_table()
235
236         self.get_table()
237
238 def set_name(self):
239     """This function is used to set the name of the user"""
240
241     with open('./table/settings/name.pickle', 'wb') as f:
242         pickle.dump(self.name, f, pickle.HIGHEST_PROTOCOL)
243
244 def get_name(self):
245     """Get the name of the user
246
247     :return: The name of the user
248     :rtype: list
249     """
250
251     try:
252         with open('./table/settings/name.pickle', 'rb') as f:
253             # The protocol version used is detected automatically, so we do not
254             # have to specify it.
255             self.name = pickle.load(f)
256
257             return self.name
258     except FileNotFoundError:
259         self.create()
260         self.set_table()
261         self.set_name()
262
263         self.get_colonne_heure()
264
265
266 if __name__ == '__main__':
267     pass
268

```