

```

1  """
2  This module is in charge of the user interface, i.e. it
   shows the right text in the chosen language.
3  """
4
5  import screen_constants
6
7
8  def show_start_ask_language():
9      """
10     asks the language the user wants to use at the
       beginning: french or english
11     ' f ' for french
12     ' e ' for english
13     :return: char
14     """
15     return input(screen_constants.START_ASK_LANGUAGE)
16
17
18  def show_language_settings():
19      """
20     asks the language the user wants to use in the program
       : french or english
21     ' f ' for french
22     ' e ' for english
23     :return: char
24     """
25     return input(screen_constants.LANGUAGE_SETTINGS)
26
27
28  def show_main_menu(english):
29      """
30     asks the user if he wants to use the fonction
       encryption or decryption of the program
31     'c' for encryption
32     'd' for decryption
33     's' for going to language settings
34     'q' to quit the program
35     :param english: boolean
36     True if chosen language is english
37     False if chosen language is french
38     :return: char
39     """
40     # language chosen is english
41     if english:

```

```

42         return input(screen_constants.ENGLISH_MAIN_MENU)
43     # language chosen is french
44     elif english is False:
45         return input(screen_constants.FRENCH_MAIN_MENU)
46     else:
47         return "this should never happen"
48
49
50 def show_principles(english, encrypting):
51     """
52     asks what type of method the user wants to use to
    encrypt or decrypt his text
53     'c' for Caesar's cypher
54     'v' for Vigenere's cypher
55     'e' for the Enigma machine
56     'm' to go back to the main menu
57     :param english: boolean
58         True if chosen language is english
59         False if chosen language is french
60     :param encrypting: boolean
61         True if the user wants to encrypt his text
62         False if he wants to decrypt his text
63     :return: char
64     """
65     # language chosen is english
66     if english:
67         if encrypting:
68             return input(screen_constants.
    ENGLISH_PRINCIPLES_ENCRYPTING)
69         if encrypting is False:
70             return input(screen_constants.
    ENGLISH_PRINCIPLES_DECRYPTING)
71     # language chosen is french
72     elif english is False:
73         if encrypting:
74             return input(screen_constants.
    ENGLISH_PRINCIPLES_ENCRYPTING)
75         if encrypting is False:
76             return input(screen_constants.
    ENGLISH_PRINCIPLES_DECRYPTING)
77     else:
78         return "this should never happen"
79
80
81 def show_ask_key(english, principle):

```

```

82     """
83     asks the key the user wants to use for the method his
      has chosen
84     key = char
85     ' m ' to go back to the main menu
86     :param principle:
87     :param english: boolean
88     True if chosen language is english
89     False if chosen language is french
90     :return: char
91     """
92     # language chosen is english
93     if english:
94         if principle == 'c':
95             return input(screen_constants.
ENGLISH_ASK_KEY_CAESAR)
96         elif principle == 'v':
97             return input(screen_constants.
ENGLISH_ASK_KEY_VIGENERE)
98         elif principle == 'e':
99             return input(screen_constants.
ENGLISH_ASK_KEY_ENIGMA)
100         else:
101             return "this should never happen"
102     # language chosen is french
103     elif english is False:
104         if principle == 'c':
105             return input(screen_constants.
FRENCH_ASK_KEY_CAESAR)
106         elif principle == 'v':
107             return input(screen_constants.
FRENCH_ASK_KEY_VIGENERE)
108         elif principle == 'e':
109             return input(screen_constants.
FRENCH_ASK_KEY_ENIGMA)
110         else:
111             return "this should never happen"
112
113
114 def show_ask_text(english):
115     """
116     asks the text that the user wants to decrypt/encrypt
117     ' m ' to go back to the main menu
118     :param english: boolean
119     True if chosen language is english

```

```
120     False if chosen language is french
121     :return: char
122     """
123     # language chosen is english
124     if english:
125         return input(screen_constants.ENGLISH_ASK_TEXT)
126     # language chosen is french
127     elif english is False:
128         return input(screen_constants.FRENCH_ASK_TEXT)
129     else:
130         return "this should never happen"
131
132
133 def show_treated_text(english, encrypting, text):
134     """
135     shows the decrypted/encrypted text
136     :param text: string
137     the user's en-/decrypted text
138     :param english: boolean
139     True if chosen language is english
140     False if chosen language is french
141     :param encrypting: boolean
142     True if the user wants to encrypt his text
143     False if he wants to decrypt his text
144     :return: char
145     """
146     # language chosen is english
147     if english:
148         if encrypting:
149             print(screen_constants.ENGLISH_ENCRYPTED_TEXT
150             + "\n" + text) # print encrypted text
151             elif encrypting is False:
152                 print(screen_constants.ENGLISH_DECRYPTED_TEXT
153                 + "\n" + text) # print decrypted text
154                 # tell user what to do in order to continue
155                 return input(screen_constants.ENGLISH_CONTINUE)
156         # language chosen is french
157         if english is False:
158             if encrypting:
159                 print(screen_constants.FRENCH_ENCRYPTED_TEXT
160                 + "\n" + text) # print encrypted text
161                 elif encrypting is False:
162                     print(screen_constants.FRENCH_DECRYPTED_TEXT
163                     + "\n" + text) # print decrypted text
164                     # tell user what to do in order to continue
```

```
161         return input(screen_constants.FRENCH_CONTINUE)
162     else:
163         return "this should never happen"
164
165
166 def show_help_principles(english):
167     """
168     shows the help screen which give explanations on the
169     different encryption methods
170     :param english: boolean
171     True if chosen language is english
172     False if chosen language is french
173     :return: char
174     """
175     # language chosen is english
176     if english:
177         input(screen_constants.ENGLISH_HELP_PRINCIPLES)
178     # language chosen is french
179     elif english is False:
180         input(screen_constants.FRENCH_HELP_PRINCIPLES)
181     else:
182         return "this should never happen"
183
184 def show_quit_message(english):
185     """
186     shows the exit message of the program
187     :param english: boolean
188     True if chosen language is english
189     False if chosen language is french
190     :return: char
191     """
192     # language chosen is english
193     if english:
194         print(screen_constants.ENGLISH_QUIT_MESSAGE)
195     # language chosen is french
196     elif english is False:
197         print(screen_constants.FRENCH_QUIT_MESSAGE)
198     else:
199         return "this should never happen"
200
```