Rapport de l'application : Quiz Generator

Explication *générale* du fonctionnement :

L'application à pour but de Génèrer un **quiz** aléatoire à partir d'un fichier (d'extension .txt) donné, ce fichier contient les questions avec leurs différentes options et suit une **syntaxe précise** que nous expliquerons plus tard . À la fin du quiz, l'utilisateur reçoit son **résultat** ainsi que la **correction** du quiz.

Démonstration de fonctionnement par un exemple :

In [1]:
```python
# voici le code source de l'application :
```

In [2]:
```python
import random  # shuffling the order of the questions and the order of the options of the questions.
import os  # check the validity of the link provided by the user

# -----
def convert(string):  # transforms a string into a list of characters
    string_list = list(string.split(","))
    return string_list

# -----
def file_2_dictionary_and_options_array(pathFile):
    tuple_array_modified, dictionary, options_array = [], {}, []
    array_lines = [line.strip() for line in open(pathFile, 'r')]  # List that contains the lines of the file -> *path*
    non_empty_array_lines = [line for line in array_lines if line.strip() != ""]

    # Dividing the list : *array_lines* into tuples of length two :
    # *tuple_array* -> tuple_array = [ (question, options ), ... ] = [ (question, [option_1, option_2, ...] ), ... ]
    tuple_array = [(non_empty_array_lines[i], non_empty_array_lines[i + 1]) for i in
                   range(0, len(non_empty_array_lines) - 1, 2)]

    for tuple_element in tuple_array:# Convert the second element of each tuple to an array: tuple_array -> tuple_array_modified
        tuple_array_modified.append((tuple_element[0], tuple_element[1].split(",")))

    for tuple_element in tuple_array_modified:  # Dictionary filling : *dictionary*  -> key : tuple_element[0] : question
        dictionary[tuple_element[0]] = tuple_element[1][
            0]  # -> value : tuple_element[1][0] : correct option of the question

    for tuple_element in tuple_array:  # List filling : *options_array*
        temp = convert(tuple_element[1])  # temp = [ option_1, option_2, .... , option_k ]
        for i in range(len(temp)):
            temp[i] = temp[i].strip()
        temp.append(tuple_element[0])  # temp = [ option_1, option_2, .... , option_k , question ]
        options_array.append(
            temp)  # options_array = [ temp, ... ] = [ [ option_1, option_2, .... , option_k , question ], ... ]

    return dictionary, options_array

# -----
def shuffle_dictionary(dictionary):  # shuffle the given dictionary
    keys = list(dictionary.keys())
    random.shuffle(keys)  # shuffle the keys
    return dict([(key, dictionary[key]) for key in keys])  # returning a new dictionary with the new shuffled keys

# -----
def get_quiz_path():
    print("\n" +  # ASCII ART LOGO : Quiz Generator
          "   ____          _           _____                                           _                   \n" +
          "  / __ \\         (_)         / ____|                                         | |                  \n" +
          " | |  | |  _   _   _   ____  | |  __    ___   _ __     ___   _ __    __ _  | |_     ___    _ __ \n" +
          " | |  | | | | | | | | |_  /  | | |_ |  / _ \\ | '_ \\   / _ \\ | '__|  / _` | | __|  / _ \\  | '__|\n" +
          " | |__| | | |_| | | |  / /   | |__| | |  __/ | | | | |  __/ | |    | (_| | | |_  | (_) | | |   \n" +
          "  \\___\\_\\  \\__,_| |_| /___|   \\_____|  \\___| |_| |_|  \\___| |_|     \\__,_|  \\__|  \\___/  |_|   \n" +
          "                                                                      @saad_labriji\n")
    print("[SYSTEM] * user can provide uppercase or lowercase choices\n")
    print("[SYSTEM] * Enter the link of the .txt file containing the quiz : \n", end="")
    path_file = input("[USER] : ")
    while os.stat(path_file).st_size == 0:
        print("[SYSTEM] *** WARNING ***  Empty file !, try again ... \n", end="")
        path_file = input("[USER] : ")
    print()
    return path_file

# -----
def new_game(dictionary, options_array):
    cpt, score = 1, 0
    user_guesses, correct_guesses, random_options_dictionary = [], [], {}
    if user_choice() == 1:  # Random order of questions
        dictionary = shuffle_dictionary(dictionary)

    for question in dictionary.keys():
        print("- Question " + str(cpt) + "/" + str(len(dictionary)) + " : ",
              end="")  # str(len(dictionary)) -> total number of Questions
        print(question)
        for options in options_array:
            if options[-1] == question:
                random_options_dictionary = random_options(options[:-1], split("ABCDEFGHIJKLMNOPQRSTUVWXYZ"))
                # 1 option -> A. option 1 , ... , 2 options -> A. option 1 , B. option 2  ...
                print("\n--------------------------------------------")
                options_selection("ABCDEFGHIJKLMNOPQRSTUVWXYZ", len(options[:-1]))
        guess = input("[USER] : ").upper()
        while guess not in random_options_dictionary:  # loop that ensures the validity of the user's choice.
            print("[SYSTEM] *** WARNING ***  invalid option ! , TRY AGAIN ... \n", end="")
            guess = input("[USER] : ").upper()
        user_guesses.append(random_options_dictionary[guess])  # Saving user results
        correct_guesses.append(dictionary.get(question))  # Saving correct results
        score += check_answer(random_options_dictionary[guess],
                              dictionary.get(question))  # Checking the result and updating the score
        print("--------------------------------------------" + "\n")
        cpt += 1  # cpt -> question index
    display_results1(user_guesses, correct_guesses, score, len(dictionary))

# -----
def user_choice():
    print("[SYSTEM] * Do you want to activate : random order of the questions ? (yes/no) \n", end="")
    choice_random_questions = input("[USER] : ")
    while choice_random_questions.lower() not in ("yes", "no"):  # loop that ensures the validity of the user's choice.
        print("[SYSTEM] *** WARNING ***  you SHOULD type yes OR no ! \n", end="")
        choice_random_questions = input("[USER] : ")
    if choice_random_questions.lower() == "yes":
        choice_random_questions = 1
    else:
        choice_random_questions = 0
    print()
    return choice_random_questions

# -----
def check_answer(guess, answer):
    guess = guess.strip().upper()
    answer = answer.strip().upper()
    print("[SYSTEM] *", end="")
    if guess == answer:
        print(" Correct! ( you get +1 point )")
        return 1
    else:
        print(" False! ( you get 0 points :/ ) ")
        return 0

# ----
def bloc_len(list1, list2):  # list1 and list2 are lists of strings
    max = 0
    for e in list1 + list2:
        if len(e) > max:
            max = len(e)
    return max  # ( length of longest element in  list1 ∪ list2 )

# ----
def create_bloc(string, n):  # exp : create_bloc( "test_string", 5) == "| test_string   |"
    if len(string) == n:
        return "| " + string + " |"
    elif len(string) == n - 1:
        return "| " + string + "  |"
    str1 = "| " + string
    for i in range(n - len(str1) + 2):
        str1 += " "
    str1 += " |"
    return str1

# ----
def hyphens_maker(n):  # exp : hyphens_maker(4) == "----"
    hyphen = ""
    for j in range(n):
        hyphen += "-"
    return hyphen

# ----
def hyphens_line(n):  # exp : hyphens_line(5) == "|------:||:-----:||:------|"
    print("|" + hyphens_maker(n + 1) + ":||:" + hyphens_maker(n) + ":||:" + hyphens_maker(n + 1) + "|")

# ----
def hyphens_last_line(n):  # exp : hyphens_last_line(6) == "|-----------------:||:-------|"
    print("|" + hyphens_maker(2 * n + 5) + ":||:" + hyphens_maker(n + 1) + "|")

# ----
def result_calculator(str1, str2):
    str1 = str1.strip().upper()
    str2 = str2.strip().upper()
    if str1 == str2:
        return "+1 point"
    else:
        return "+0 point"

# ----
def display_results1(list1, list2, score, number_questions):
    n = max(bloc_len(list1, list2) + 4, 22)
    print("[SYSTEM] * Here is your results : \n")
    print(create_bloc(" ", n)[1:-1] + " |", end="")
    hyphens_line(n)
    print(create_bloc(" ", n)[1:-1] + " |", end="")
    print(create_bloc("User guesses ", n) + create_bloc("Correct answers  ", n) + create_bloc("Points ", n))
    print("| " + hyphens_maker(n) + " |", end="")
    hyphens_line(n)
    for i in range(len(list1)):
        print(create_bloc(" Question N°" + str(i + 1), n), end="")
        print(
            create_bloc(list1[i], n) + create_bloc(list2[i], n) + create_bloc(result_calculator(list1[i], list2[i]), n))
    print("| " + hyphens_maker(n) + " |", end="")
    hyphens_last_line(n)
    print(create_bloc(" ", n)[1:-1] + " |", end="")
    print(create_bloc("Final result", 2 * n + 4) + create_bloc(str(score) + " points", n))
    print(create_bloc(" ", n)[1:-1] + " |", end="")
    print(create_bloc("Percentage of success", 2 * n + 4) + create_bloc(
        "{:.2f}".format(score / number_questions * 100) + " %", n))
    print(create_bloc(" ", n)[1:-1] + " |", end="")
    hyphens_last_line(n)

# ----
def random_options(options_list, list_numbers):
    random_options_dictionary = {}
    n = len(options_list)
    while len(options_list) != 0:
        random_option = random.choice(options_list)
        print("      " + list_numbers[n - len(options_list)] + ". " + random_option)
        random_options_dictionary[list_numbers[n - len(options_list)]] = random_option
        options_list.remove(random_option)
    return random_options_dictionary

# -----
def split(word):  # word to list of characters
    return [char for char in word]

# -----
def select_options(string, n):  # prints the n first characters of the word : *word*
    string = string[:n]
    for e in string:
        print(e)

# ----------
def options_selection(string, n):  # exp : options_selection("ABCDEF", 4) == "[SYSTEM] * Select (A, B, C, or D) :"
    if n == 2:
        print("[SYSTEM] * Select (" + string[0] + " or " + string[1] + ") :\n", end="")
    else:
        string = string[:n]
        print("[SYSTEM] * Select (", end="")
        for e in string[:-1]:
            print(e, end=", ")
        print("or " + string[-1] + ") :\n", end="")


                                                  # coded by @saad_labriji
                                                  # in 23 may 2021
```

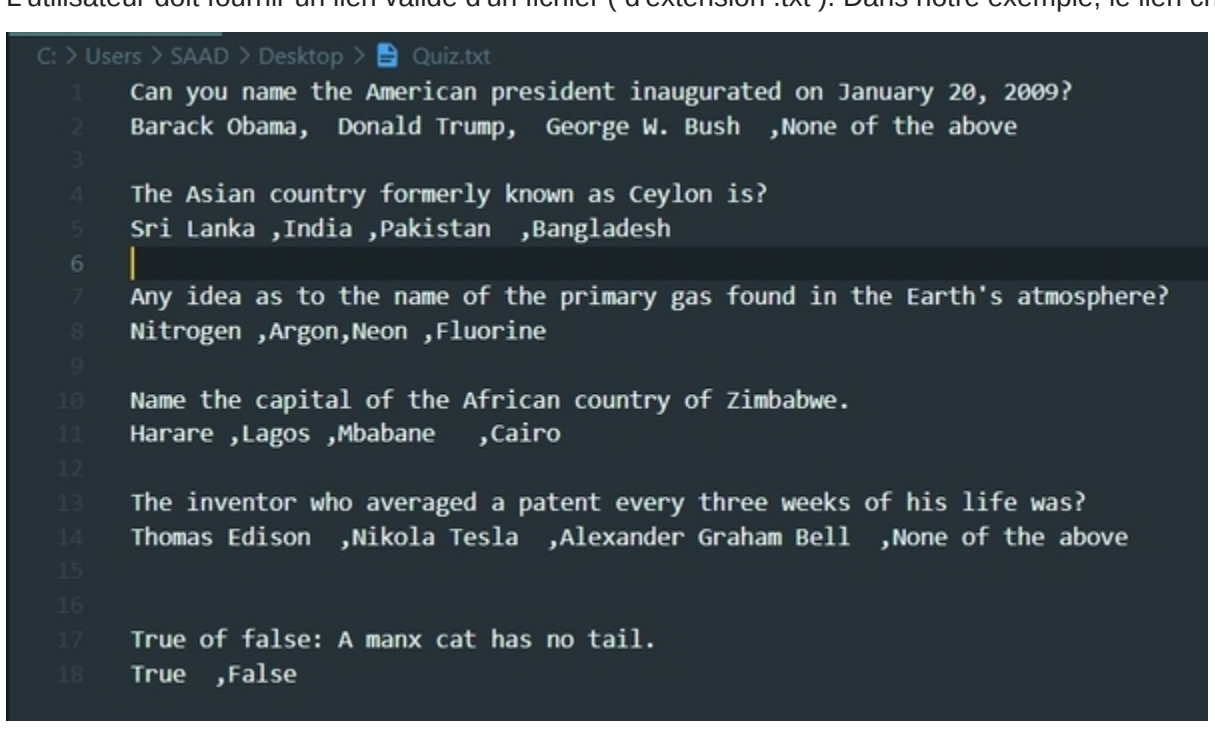
In [3]:
```python
# Lancement de l'application : ( suite du code source )
```

In [4]:
```python
try:
    path = get_quiz_path()
    file_tuple = file_2_dictionary_and_options_array(path)
    new_game(file_tuple[0], file_tuple[1]) # ---------- Run the Quiz :

except Exception as exception: # Exception handling :
    print("\n[SYSTEM] *** WARNING ***  something went wrong :( ")
    print("         *** ", end="")
    print(exception)
```

```
   ____          _           _____                                           _
  / __ \        (_)         / ____|                                         | |
 | |  | |  _   _   _   ____  | |  __    ___   _ __     ___   _ __    __ _  | |_     ___    _ __
 | |  | | | | | | | | |_  /  | | |_ |  / _ \ | '_ \   / _ \ | '__|  / _` | | __|  / _ \  | '__|
 | |__| | | |_| | | |  / /   | |__| | |  __/ | | | | |  __/ | |    | (_| | | |_  | (_) | | |
  \___\_\  \__,_| |_| /___|   \_____|  \___| |_| |_|  \___| |_|     \__,_|  \__|  \___/  |_|
                                                                      @saad_labriji

[SYSTEM] * user can provide uppercase or lowercase choices

[SYSTEM] * Enter the link of the .txt file containing the quiz :
[USER] : C:\Users\SAAD\Desktop\Quiz.txt

[SYSTEM] * Do you want to activate : random order of the questions ? (yes/no)
[USER] : YeS

- Question 1/6 : The inventor who averaged a patent every three weeks of his life was?
      A. Thomas Edison
      B. Nikola Tesla
      C. None of the above
      D. Alexander Graham Bell

--------------------------------------------
[SYSTEM] * Select (A, B, C, or D) :
[USER] : a
[SYSTEM] * Correct! ( you get +1 point )
--------------------------------------------

- Question 2/6 : True of false: A manx cat has no tail.
      A. False
      B. True

--------------------------------------------
[SYSTEM] * Select (A or B) :
[USER] : b
[SYSTEM] * Correct! ( you get +1 point )
--------------------------------------------

- Question 3/6 : Any idea as to the name of the primary gas found in the Earth's atmosphere?
      A. Nitrogen
      B. Neon
      C. Fluorine
      D. Argon

--------------------------------------------
[SYSTEM] * Select (A, B, C, or D) :
[USER] : f
[SYSTEM] *** WARNING ***  invalid option ! , TRY AGAIN ...
[USER] : D
[SYSTEM] * False! ( you get 0 points :/ )
--------------------------------------------

- Question 4/6 : The Asian country formerly known as Ceylon is?
      A. Sri Lanka
      B. India
      C. Bangladesh
      D. Pakistan

--------------------------------------------
[SYSTEM] * Select (A, B, C, or D) :
[USER] : a
[SYSTEM] * Correct! ( you get +1 point )
--------------------------------------------

- Question 5/6 : Name the capital of the African country of Zimbabwe.
      A. Cairo
      B. Mbabane
      C. Harare
      D. Lagos

--------------------------------------------
[SYSTEM] * Select (A, B, C, or D) :
[USER] : b
[SYSTEM] * False! ( you get 0 points :/ )
--------------------------------------------

- Question 6/6 : Can you name the American president inaugurated on January 20, 2009?
      A. Barack Obama
      B. Donald Trump
      C. George W. Bush
      D. None of the above

--------------------------------------------
[SYSTEM] * Select (A, B, C, or D) :
[USER] : e
[SYSTEM] *** WARNING ***  invalid option ! , TRY AGAIN ...
[USER] : D
[SYSTEM] * False! ( you get 0 points :/ )
--------------------------------------------

[SYSTEM] * Here is your results :

                         ||----------------------:||:---------------------:||:----------------------|
                         || User guesses          || Correct answers       || Points               |
| ---------------------- ||----------------------:||:---------------------:||:----------------------|
|  Question N°1          || Thomas Edison         || Thomas Edison         || +1 point             |
|  Question N°2          || True                  || True                  || +1 point             |
|  Question N°3          || Argon                 || Nitrogen              || +0 point             |
|  Question N°4          || Sri Lanka             || Sri Lanka             || +1 point             |
|  Question N°5          || Mbabane               || Harare                || +0 point             |
|  Question N°6          || None of the above     || Barack Obama          || +0 point             |
| ---------------------- ||----------------------------------------------:||:----------------------|
                         || Final result                                  || 3 points             |
                         || Percentage of success                         || 50.00 %              |
                         ||----------------------------------------------:||:----------------------|
```

Explication *détaillée* du fonctionnement sur l'exemple précédent :

L'utilisateur doit fournir un lien valide d'un fichier (d'extension .txt). Dans notre exemple, le lien choisis est : **C:\Users\SAAD\Desktop\Quiz.txt** , le contenu du fichier est :



C: > Users > SAAD > Desktop > Quiz.txt

```
Can you name the American president inaugurated on January 20, 2009?
Barack Obama,  Donald Trump,  George W. Bush  ,None of the above

The Asian country formerly known as Ceylon is?
Sri Lanka ,India ,Pakistan  ,Bangladesh

Any idea as to the name of the primary gas found in the Earth's atmosphere?
Nitrogen ,Argon,Neon ,Fluorine

Name the capital of the African country of Zimbabwe.
Harare ,Lagos ,Mbabane   ,Cairo

The inventor who averaged a patent every three weeks of his life was?
Thomas Edison  ,Nikola Tesla  ,Alexander Graham Bell  ,None of the above


True of false: A manx cat has no tail.
True  ,False
```

Le quiz ci-dessus est composé de 6 questions, chaque question est directement suivie de leurs options. La première option pour chaque question est celle considérer correcte, l'application compare le choix de l'utilisateur parmi les options avec la première option.

En prenant la question suivante par exemple :

```
- Question 5/6 : The inventor who averaged a patent every three weeks of his life was?
      A. Nikola Tesla
      B. Alexander Graham Bell
      C. None of the above
      D. Thomas Edison
```

La réponse considérée comme correcte à cette question est : **Thomas Edison** (car c'est la premiére option de la qusetion)

À la fin du quiz, l'utilisateur reçoit un tableau récapitulatif qui contient l'historique des réponses, son score et son pourcentage de réussite, ainsi que la correction du quiz.

En prenant l'exemple précédent, le tableau récapitulatif fourni est:

```
                         ||----------------------:||:---------------------:||:----------------------|
                         || User guesses          || Correct answers       || Points               |
| ---------------------- ||----------------------:||:---------------------:||:----------------------|
|  Question N°1          || Thomas Edison         || Thomas Edison         || +1 point             |
|  Question N°2          || True                  || True                  || +1 point             |
|  Question N°3          || Argon                 || Nitrogen              || +0 point             |
|  Question N°4          || Sri Lanka             || Sri Lanka             || +1 point             |
|  Question N°5          || Mbabane               || Harare                || +0 point             |
|  Question N°6          || None of the above     || Barack Obama          || +0 point             |
| ---------------------- ||----------------------------------------------:||:----------------------|
                         || Final result                                  || 3 points             |
                         || Percentage of success                         || 50.00 %              |
                         ||----------------------------------------------:||:----------------------|
```

Syntaxe du fichier .txt à fournir à l'application :

Voici les spécifications que le fichier fourni doit respecter pour que le programme fonctionne normalement :

- Le fichier doit avoir une extension .txt
- Chaque question doit couvrir une seule ligne (la seule spécification à laquelle la question doit répondre).
- Chaque question doit être suivie de leurs options (toutes les options d'une question couvrent une seule ligne).
  - La première option est considérée comme **correcte** par le programme.
  - Les options doivent être séparées par des virgules (les options ne peuvent pas contenir de virgules [,])
- Le fichier fourni peut contenir des lignes vides n'importe où (les lignes vides sont traitées et supprimées automatiquement par le programme).
- Les espaces excessifs sont autorisés et sont automatiquement traitées et supprimées par le programme.

Par exemple, les options suivantes sont autorisées :

```
Sri Lanka,India,Pakistan,Bangladesh                   # pas d'espaces excessifs.
Sri Lanka    ,India    ,Pakistan   ,Bangladesh        # espaces excessifs à droite.
    Sri Lanka,    India   ,Pakistan   ,Bangladesh     # espaces excessifs à gauche.
 Sri Lanka   ,  India    ,Pakistan  ,  Bangladesh     # espaces excessifs à gauche et à droite.
```

Remarque : comme mentionné précédemment **Sri Lanka** est l'option correcte de la question.

Cas où l'application ne fonctionne pas correctement :

voici les cas les plus susceptibles de provoquer un dysfonctionnement de l'application :

- Le lien fourni par l'utilisateur est invalide, le système envoie un message d'alerte : something went wrong :(

In [ ]:
```
[SYSTEM] * Enter the link of the .txt file containing the quiz :
[USER] : fake path

[SYSTEM] *** WARNING ***  something went wrong :(
         *** [WinError 2] Le fichier spécifié est introuvable: 'fake path'
```

- Le fichier fourni ne respecte pas la syntaxe prédéfinie (l'utilisateur reçoit du n'importe quoi)

- L'utilisateur choisit une option inexistante (le programme demande une nouvelle option).

In [ ]:
```
- Question 1/6 : The inventor who averaged a patent every three weeks of his life was?
      A. None of the above
      B. Alexander Graham Bell
      C. Thomas Edison
      D. Nikola Tesla

--------------------------------------------
[SYSTEM] * Select (A, B, C, or D) :
[USER] : h
[SYSTEM] *** WARNING ***  invalid option ! , TRY AGAIN ...
[USER] : a
[SYSTEM] * False! ( you get 0 points :/ )
--------------------------------------------
```

Application codée par @saad_labriji : (23 mai 2021) LinkedIn GitHub