



Information sur le fonctionnement du programme SAE Interface



INSTITUT UNIVERSITAIRE D'INFORMATIQUE

ANNEE 2023 - 2024

Document Rédigé par Dupuis Brian et Payen Alexis

Avant de lancer le programme merci de créer une database en SQL et de la remplir des données contenues dans le fichier, merci de vérifier si vous possédez les bibliothèques : matplotlib, Pyqt6 et psycpg2

Une fois le programme lancé il vous demandera la Database merci de rentré le nom de la Database contenant les données SQL précédemment remplir, puis de rentré votre nom d'utilisateur et enfin votre mot de passe.

Ligne 1 à 25 :

```
1 # programme développer par Dupuis Brian, Payen Alexis, Mazuier Eve et Lhote Florian
2
3 # se programme utilise les information d'une base SQL pour fonctionné il affiché le nombre de kg de CO2 produit
4 # en fonction de la distanc, pour plus d'information je vous recommande de regarde le pdf fournie avec le programme
5
6 ##### importation #####
7
8 import sys
9 from PyQt6.QtWidgets import QApplication, QWidget, QVBoxLayout, QPushButton, QHBoxLayout, QLabel, QPushButton
10 import matplotlib
11 matplotlib.use('QtAgg')
12 from PyQt6 import QtWidgets
13 from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg
14 from matplotlib.figure import Figure
15 import psychopg2
16
17 ##### connection a la base SQL #####
18
19 conn = psychopg2.connect(
20     host="localhost",
21     database=input("nom de la database : "),
22     port="5432",
23     user=input("votre nom d'utilisateur : "),
24     password=input("votre mot de passe : ")
25 )
```

Entre les lignes 1 à 15 nous avons les importations des bibliothèques nécessaires.

De la ligne 19 à 25 nous avons le programme qui permettra de se connecter à votre base SQL, il se connecte de la même façon que vous quand vous lancé un SQL Shell.

Ligne 29 à 779 :

```
1 ##### partie 1 : Les 10 plus grand aeroport #####
2
3
4 # en fonction du y on aura accès aux dictionnaire desire
5 if y==0: # il permet de mettre a zero le programme a son lancement
6     distance_avions=({})
7     CO2_avions=({})
8
9 elif y==1: # son fonctionnement et la même pour la totalité des if de cette fonction
10     cursor = conn.cursor() # on prepare a lancé des commande SQL
11     cursor.execute("SELECT distance FROM aero_1 ;") # on execute la commande SQL qui nous donne la totalité de la distance
12     distance_avions = cursor.fetchall() # on stock les données dans une variable de type dictionnaire
13
14     cursor.execute("SELECT emissions_co2/100 FROM aero_1 ;") # on execute la commande SQL qui nous donne la totalité du CO2 on divise par 100 pour transformé le gramme de Co2 en kilogramme de CO2
15     CO2_avions = cursor.fetchall() # on stock les données dans une variable de type dictionnaire
16
17 elif y==2:
18     cursor = conn.cursor() # on prepare a lancé des commande SQL
19     cursor.execute("SELECT distance FROM aero_2 ;")
20     distance_avions = cursor.fetchall()
21
22     cursor.execute("SELECT emissions_co2/100 FROM aero_2 ;")
23     CO2_avions = cursor.fetchall()
24
25 elif y==3:
26     cursor = conn.cursor() # on prepare a lancé des commande SQL
27     cursor.execute("SELECT distance FROM aero_3 ;")
28     distance_avions = cursor.fetchall()
29
30     cursor.execute("SELECT emissions_co2/100 FROM aero_3 ;")
31     CO2_avions = cursor.fetchall()
32
33 elif y==4:
34     cursor = conn.cursor() # on prepare a lancé des commande SQL
35     cursor.execute("SELECT distance FROM aero_4 ;")
36     distance_avions = cursor.fetchall()
37
38     cursor.execute("SELECT emissions_co2/100 FROM aero_4 ;")
39     CO2_avions = cursor.fetchall()
40
41 elif y==5:
42     cursor = conn.cursor() # on prepare a lancé des commande SQL
43     cursor.execute("SELECT distance FROM aero_5 ;")
44     distance_avions = cursor.fetchall()
45
46     cursor.execute("SELECT emissions_co2/100 FROM aero_5 ;")
47     CO2_avions = cursor.fetchall()
48
49 elif y==6:
50     cursor = conn.cursor() # on prepare a lancé des commande SQL
51     cursor.execute("SELECT distance FROM aero_6 ;")
52     distance_avions = cursor.fetchall()
53
54     cursor.execute("SELECT emissions_co2/100 FROM aero_6 ;")
55     CO2_avions = cursor.fetchall()
```

```

73 distance_avions = distance_avions + distance_avions_1
74
75 cursor.execute("SELECT emissions_co2/100 FROM aero_5 ;")
76 CO2_avions_2 = cursor.fetchall()
77
78 CO2_avions = CO2_avions + CO2_avions_2
79
80 cursor.execute("SELECT distance FROM aero_6 ;")
81 distance_avions_1 = cursor.fetchall()
82
83 distance_avions = distance_avions + distance_avions_1
84
85 cursor.execute("SELECT emissions_co2/100 FROM aero_6 ;")
86 CO2_avions_2 = cursor.fetchall()
87
88 CO2_avions = CO2_avions + CO2_avions_2
89
90 cursor.execute("SELECT distance FROM aero_7 ;")
91 distance_avions_1 = cursor.fetchall()
92
93 distance_avions = distance_avions + distance_avions_1
94
95 cursor.execute("SELECT emissions_co2/100 FROM aero_7 ;")
96 CO2_avions_2 = cursor.fetchall()
97
98 CO2_avions = CO2_avions + CO2_avions_2
99
100 cursor.execute("SELECT distance FROM aero_8 ;")
101 distance_avions_1 = cursor.fetchall()
102
103 distance_avions = distance_avions + distance_avions_1
104
105 cursor.execute("SELECT emissions_co2/100 FROM aero_8 ;")
106 CO2_avions_2 = cursor.fetchall()
107
108 CO2_avions = CO2_avions + CO2_avions_2
109
110 cursor.execute("SELECT distance FROM aero_9 ;")
111 distance_avions_1 = cursor.fetchall()
112
113 distance_avions = distance_avions + distance_avions_1 # CO2_avions possède maintenant la totalité des valeurs de distance en kilomètre des 9 aeroport
114
115 cursor.execute("SELECT emissions_co2/100 FROM aero_9 ;")
116 CO2_avions_2 = cursor.fetchall()
117
118 CO2_avions = CO2_avions + CO2_avions_2 # CO2_avions possède maintenant la totalité des valeurs de CO2 en Kilogramme des 9 aeroport
119
120 #####
121
122 if distance_avions==[] or CO2_avions==[]: # si l'une des 2 variable est vide alors on met les deux variable à 0
123     distance_avions=[(0,0)]
124     CO2_avions=[(0,0)]
125
126 return distance_avions,CO2_avions # on return les dictionnaires

```

cette fonction qui prend une grande place du programme sert à effectuer des commandes SQL sur python pour pouvoir récupérer les données nécessaires c'est à dire la distance qui est en km mais aussi la quantité de CO2 produit en g mais qu'on va transformé en kg en divisant par 100, si cette partie du programme est si long est dû au fait qu'on ne peut pas le raccourcir ou difficilement se problème viens du fait que chaque commande SQL est différente et du à un grand nombre de boutons qui nécessite des données se trouvant sur SQL obligeant un grand nombre de ligne, parmi tous ces if il y en a 2 qui sont plus grand que les autres car comparé aux autres qui prend les informations d'un aéroport c'est deux if qui prennent les informations de plusieurs aéroports.

En ce qui concerne le fonctionnement de la fonction quand on l'appelle elle doit recevoir un int permettant de savoir dans quel dictionnaire elle doit aller un fois cela fait il exécutera 2 commandes SQL la première qui permettra d'obtenir les données de distance en km qui sera stocké dans la variable « distance avions » la deuxième commande SQL nous donnera la quantité de CO2 en kg qui sera stocké dans la variable « CO2_avions ».

En ce qui concerne l'if à la fin est une sécurité au cas où que l'une des deux variables est vide pour éviter un plantage du logiciel

Ligne 783 - 798

```

781 ##### calculé des moyenne et de la sommes de tout les produits #####
782
783 def distance_co2(self,x): # cette fonction permet de faire les calcul pour le total et les moyennes de co2 et de distance
784     p0, p1=aerop(x) # on appelle les dictionnaire voulus
785     i = 0
786     self.text1 = 0 # on met à zero les 4 text
787     self.text2 = 0
788     self.text3 = 0
789     self.text4 = 0
790     while i<len(p1)-1: # on va calculé la moyenne et le total de tout les valeur de l'aéroport
791         p2=p0[i] # on récupère les valeurs de p0[i] pour qu'elle soit utilisable
792         p3=p1[i] # on récupère les valeurs de p1[i] pour qu'elle soit utilisable
793
794         self.text1 = self.text1+p2[0] # on additionne les valeurs de text1 avec les valeur de p2
795         self.text2 = self.text2+p3[0] # on additionne les valeurs de text2 avec les valeur de p3
796         i = i+1
797     self.text3 = self.text1 // len(p1) # une fois additionné on calcule la moyenne de CO2 qu'on stock dans text3
798     self.text4 = self.text2 // len(p1) # uneune fois additionné on calcule le moyenne de la distance qu'on stock dans text4

```

Cette fonctionne permet de calculer la moyenne de la distance et du CO2 par vol qui seront stock dans « text3 » et « text4 » mais aussi de calculer le total qui seront eu stocké dans « text1 » et « text2 » elle prend aussi une valeur en paramètre de type x.

La fonction fonctionne de cette façon ont appelé d'abord « aerop (x) » qui nous permet d'avoir les valeurs de distance et de CO2 dans 2 variables de type dictionnaire contenant des floats qui sont respectivement p1 et p0 une fois cela fait on définit i qu'on met à 0 puis on met text1 à 4 à zéro avant de lancer un while qui s'arrête une fois que i serait plus grand que la longueur totale de p1-1

Suite à cela on va utiliser 2 nouvelles variable p2 et p3, p2 va prendre la valeur de p0[i] et p3 va prendre la valeur de p1[i], puis nous allons calculer le total de CO2 et de distance attention la valeur que contient p2 et p3 sont des dictionnaires vous devez donc mettre obligatoirement mettre [0] derrière les variables si vous voulez faire des calculs avec

Enfin on calcule les fonctionne pour sa c'est text3=text1 // len(p1)

Et on fait la même avec text4 : text4=textt2 // len(p1)

Ligne 802-816

```

##### création du graphique #####
class MplCanvas(FigureCanvasQTAgg): # permet de cré Le graphique
    def __init__(self, parent=None, width=5, height=4, dpi=100):
        fig = Figure(figsize=(width, height), dpi=dpi)
        self.axes = fig.add_subplot(111)
        super(MplCanvas, self).__init__(fig)

class MainWindow(QtWidgets.QMainWindow):

    def __init__(self, x, *args, **kwargs, ): # permet de crée la courbe du graphique
        super(MainWindow, self).__init__(*args, **kwargs)
        sc = MplCanvas(self, width=5, height=4, dpi=100)

        p0, p1=aerop(x) # on appelle la fonction aerop pour qu'ils nous donne les valeurs de CO2 et de distance
        sc.axes.plot(p0, p1) # on donne les 2 valeurs pour qu'ils crée le graphique
        self.setCentralWidget(sc) # on met au centre le widget

##### programme principale #####

```

Les deux fonctionnent sont important pour que l'autre fonctionne je vais parler seulement de la fin du deuxième programme faut savoir que c'est deux fonctions permet de créer un graphique avec des coordonnées

On appelle à nouveau la fonction "aerop(x)" p0 et p1 prennent les valeurs de CO2 et de distance puis on les utilise dans "sc.axes.plot(p0, p1)" il récupère les coordonnées pour créer le graphique attention il vous faut au minimum 2 valeurs dans chaque des variables pour créer votre graphique sinon il sera vide

820-961

```
##### definition des variables #####
self.toutselect="tout sélectionner"
self.toutretire="tout retirer"

self.text1 = 0.0
self.text2 = 0.0
self.text3 = 0.0
self.text4 = 0.0

##### création des layouts #####
# la création des layouts vont de gauche à droite
self.v1layout = QVBoxLayout() # création du layout tout a gauche il contiendra les widget "france", "autres" et "sélectionnée tout"
self.v2_2_layout = QVBoxLayout() # création d'un layout contenant du texte
self.v2layout = QVBoxLayout() # création d'un layout contenant des QPushButton des 10 plus grand aeroport
self.h1_layout = QHBoxLayout() # création d'un layout qui va contenir les 3 layouts precedents
self.v2_3layout = QVBoxLayout() # création d'un layout contenant du text
self.v2_1layout = QVBoxLayout() # création d'un layout contenant des QPushButton des aeroport français (58 QPushButton)
# mais aussi tout en haut le bouton "sélectionné tout" et tout en bas "suivant qui permettra de changer de page"
self.h1layout = QHBoxLayout() # création d'un layout Les 2 layouts precedent
self.v3layout = QVBoxLayout() # création d'un layout contenant du text
self.h2layout = QHBoxLayout() # création du layout principale
self.h3layout = QHBoxLayout() # création d'un layout contenant le graphique
self.setLayout(self.h2layout) # on définit h2layout en layout principale
```

Dans cette partie on s'attaque au programme principale on va premièrement créer nos premières variables puis nos Layout si vous voulez savoir à quoi servent les layouts il vous suffit de lire les commentaires

```
self.pager=1 # cette variable permet de savoir si on doit cacher ou non les QPushButton des aeroport étrangers
self.autre_pager=1 # cette variable permet de savoir si on doit cacher ou non les QPushButton des aeroport français
self.france_pager=1 # cette variable permet de savoir si on doit cacher ou non les QPushButton des aeroport français
```

cette partie sera consacré à la création de la totalité des 70 QPushButton qui serviront seront utilisés dans 3 layouts différents le layout "France", "autres", "sélectionné tout", le deuxième layout contiendra tous les widget aéroport mais aussi le widget "sélectionné tout" ou autrement nommé "aéroport 11" en fin le troisième contiendra tout en haut le widget "tout sélectionné" puis la totalité de 58 aéroport français qui seront divisés en 3 pages qu'on pourra jongler grâce au bouton "suivant" qui se trouve en bas de la page

```
self.show() # on affiche la fenetre
```

Dans cette dernière partie on définit les 3 variables qui nous permettront de savoir si on doit cacher des éléments et si oui qu'elles éléments on les montrés, puis on crée 6 lignes contenant du texte puis on appelle la fonction init () qui va initialiser l'interface puis on définit la taille de la fenêtre et de nommée interface puis on finit par afficher la fenêtre

963 – 1246


```

1193 def change_aeroport52(self) : # cette fonction appelle la fonction change_aeroport mais en donnant la valeur du dictionnaire voulu
1194     self.change_aeroport(52)
1195
1196 def change_aeroport53(self) : # cette fonction appelle la fonction change_aeroport mais en donnant la valeur du dictionnaire voulu
1197     self.change_aeroport(53)
1198
1199 def change_aeroport54(self) : # cette fonction appelle la fonction change_aeroport mais en donnant la valeur du dictionnaire voulu
1200     self.change_aeroport(54)
1201
1202 def change_aeroport55(self) : # cette fonction appelle la fonction change_aeroport mais en donnant la valeur du dictionnaire voulu
1203     self.change_aeroport(55)
1204
1205 def change_aeroport56(self) : # cette fonction appelle la fonction change_aeroport mais en donnant la valeur du dictionnaire voulu
1206     self.change_aeroport(56)
1207
1208 def change_aeroport57(self) : # cette fonction appelle la fonction change_aeroport mais en donnant la valeur du dictionnaire voulu
1209     self.change_aeroport(57)
1210
1211 def change_aeroport58(self) : # cette fonction appelle la fonction change_aeroport mais en donnant la valeur du dictionnaire voulu
1212     self.change_aeroport(58)
1213
1214 def change_aeroport59(self) : # cette fonction appelle la fonction change_aeroport mais en donnant la valeur du dictionnaire voulu
1215     self.change_aeroport(59)
1216
1217 def change_aeroport60(self) : # cette fonction appelle la fonction change_aeroport mais en donnant la valeur du dictionnaire voulu
1218     self.change_aeroport(60)
1219
1220 def change_aeroport61(self) : # cette fonction appelle la fonction change_aeroport mais en donnant la valeur du dictionnaire voulu
1221     self.change_aeroport(61)
1222
1223 def change_aeroport62(self) : # cette fonction appelle la fonction change_aeroport mais en donnant la valeur du dictionnaire voulu
1224     self.change_aeroport(62)
1225
1226 def change_aeroport63(self) : # cette fonction appelle la fonction change_aeroport mais en donnant la valeur du dictionnaire voulu
1227     self.change_aeroport(63)
1228
1229 def change_aeroport64(self) : # cette fonction appelle la fonction change_aeroport mais en donnant la valeur du dictionnaire voulu
1230     self.change_aeroport(64)
1231
1232 def change_aeroport65(self) : # cette fonction appelle la fonction change_aeroport mais en donnant la valeur du dictionnaire voulu
1233     self.change_aeroport(65)
1234
1235 def change_aeroport66(self) : # cette fonction appelle la fonction change_aeroport mais en donnant la valeur du dictionnaire voulu
1236     self.change_aeroport(66)
1237
1238 def change_aeroport67(self) : # cette fonction appelle la fonction change_aeroport mais en donnant la valeur du dictionnaire voulu
1239     self.change_aeroport(67)
1240
1241 def change_aeroport68(self) : # cette fonction appelle la fonction change_aeroport mais en donnant la valeur du dictionnaire voulu
1242     self.change_aeroport(68)
1243
1244 def change_aeroport69(self) : # cette fonction appelle la fonction change_aeroport mais en donnant la valeur du dictionnaire voulu
1245     self.change_aeroport(69)
1246
1247
1248

```

La première partie consiste à des "signal and slots" pour chaque bouton, il s'avère qu'on ne peut pas passer des variables en paramètre de se faite j'ai dû appeler une autres fonctionnent qui a pour but d'appeler la bonne fonction en passant en paramètre les valeurs nécessaires

1249-1275

```

1249 def change_aeroport(self,k) : # cette fonction sert a change Les valeurs du text mais aussi changé le graphique
1250
1251     self.v3layout.removeWidget(self.ligne1) # on retire le widget ligne 1
1252     self.v3layout.removeWidget(self.ligne2) # on retire le widget ligne 2
1253     self.v3layout.removeWidget(self.ligne3) # on retire le widget ligne 3
1254     self.v3layout.removeWidget(self.ligne4) # on retire le widget ligne 4
1255
1256     distance_co2(self,k) # on appelle la fonction distance_co2 pour qui va changé les valeur de text 1, 2, 3 et 4
1257
1258     # on remet les widget ligne 1,2,3 et 4
1259     self.ligne1 = QLabel("quantité de CO2 émit en total en kilograme : " + str(self.text1))
1260     self.ligne2 = QLabel("distance parcouru en total : " + str(self.text2))
1261     self.ligne3 = QLabel("quantité de CO2 émit en moyenne par vol en kilograme : " + str(self.text3))
1262     self.ligne4 = QLabel("distance en moyenne parcouru en moyenne : " + str(self.text4))
1263
1264     self.v3layout.addWidget(self.ligne1)
1265     self.v3layout.addWidget(self.ligne2)
1266     self.v3layout.addWidget(self.ligne3)
1267     self.v3layout.addWidget(self.ligne4)
1268
1269
1270     self.h3layout.removeWidget(self.w) # on enlève le widget w qui est le widget du graphique
1271
1272
1273     self.w = MainWindow(k) # on appelle la fonction mainwindow pour qu'elle nous crée le nouveau graphique avec les nouvelles
1274
1275     self.h3layout.addWidget(self.w) # on ajout a nouvelle le widget w

```

Cette fonction va remettre à jour les widgets concernés, on va d'abord enlever les 4 widget ligne qui vont de 1 à 4

Puis on appelle la fonction distance_CO2 qui va calculer les nouvelles valeurs des variables text1 à 4 puis on recrée les widgets ligne1 à 4 avant de les ajouter au layout puis on refait la même chose mais cette fois avec le graphique

Lignes 1279 – 1430


```

1279  def init(self):
1280
1281      # on ajoute Les widget dans Le v1layout
1282
1283      self.v1layout.addWidget(self.France)
1284      self.v1layout.addWidget(self.Autre)
1285      self.v1layout.addWidget(self.tout)
1286
1287      # on ajoute Le layout : v1layout dans Le Layout : h1_1layout
1288
1289      self.h1_1layout.addLayout(self.v1layout)
1290
1291      # on ajoute Les widget dans Le v2_2_2layout
1292
1293      self.v2_2_2layout.addWidget(self.ligne5)
1294
1295      # on ajoute Le layout : v2_2_2Layout dans Le Layout : h1_1layout
1296
1297      self.h1_1layout.addLayout(self.v2_2_2layout)
1298
1299      # on ajoute Les widget dans Le v2layout
1300
1301      self.v2layout.addWidget(self.aeroport1)
1302      self.v2layout.addWidget(self.aeroport2)
1303      self.v2layout.addWidget(self.aeroport3)
1304      self.v2layout.addWidget(self.aeroport4)
1305      self.v2layout.addWidget(self.aeroport5)
1306      self.v2layout.addWidget(self.aeroport6)
1307      self.v2layout.addWidget(self.aeroport7)
1308      self.v2layout.addWidget(self.aeroport8)
1309      self.v2layout.addWidget(self.aeroport9)
1310      self.v2layout.addWidget(self.aeroport11)
1311
1312      # on ajoute Le layout : v2layout dans Le Layout : h1_1layout
1313
1314      self.h1_1layout.addLayout(self.v2layout)
1315
1316      # on ajoute Les widget dans Le v2_2_3layout
1317
1318      self.v2_2_3layout.addWidget(self.ligne6)
1319
1320      # on ajoute Le layout : v2_2_3layout dans Le Layout : h1_1layout
1321
1322      self.h1_1layout.addLayout(self.v2_2_3layout)

```



```

1384     self.v2_1layout.addWidget(self.Montluçon_Guérét)
1385     self.v2_1layout.addWidget(self.Dole_Jura)
1386     self.v2_1layout.addWidget(self.Lorient_Bretagne_Sud)
1387     self.v2_1layout.addWidget(self.suivant)
1388
1389     # on ajoute Le Layout : v2_1layout dans Le layout : h1_1layout
1390
1391     self.h1_1layout.addLayout(self.v2_1layout)
1392
1393     # on ajoute Le Layout : h1_1layout dans Le layout : h1layout
1394
1395     self.h1layout.addLayout(self.h1_1layout)
1396
1397     # on ajoute Les widget dans Le v3layout
1398
1399     self.v3layout.addWidget(self.ligne1)
1400     self.v3layout.addWidget(self.ligne2)
1401     self.v3layout.addWidget(self.ligne3)
1402     self.v3layout.addWidget(self.ligne4)
1403
1404     # on ajoute Le Layout : h1layout dans Le layout : h2layout
1405
1406     self.h2layout.addLayout(self.h1layout)
1407
1408     # on ajoute Le Layout : v3layout dans Le layout : h2layout
1409
1410     self.h2layout.addLayout(self.v3layout)
1411
1412     # on crée une variable w qui contient graphique
1413
1414     self.w = MainWindow(0)
1415
1416     # on ajoute Le graphique dans Le h3layout
1417
1418     self.h3layout.addWidget(self.w)
1419
1420     # on ajoute Le Layout : h3layout dans Le layout : h2layout
1421
1422     self.h2layout.addLayout(self.h3layout)
1423
1424     # on appelle la fonction changement_autres qui va caché les widgets et le texte des aeroport non français
1425
1426     self.changement_autres()
1427
1428     # on appelle la fonction changement_france qui va caché les widgets et le texte des aeroport français
1429
1430     self.changement_france()

```

dans cette partie on va juste ajouter le widget a leurs layout puis ajouter les sous layout au layout plus grand pour au final tous les ressemblés dans le layout "h2layout", avant d'appeler deux fonction qui sont "changement_autres" et "changement_france" qui vont cacher tous sauf les widget "france", "autres", "tout" et le graphique.

```

1433     def change_page(self): # sa permet de navigé entre Les différentes pagesdes aeroport français
1434         self.page=self.page+1
1435         if self.page==4: # si page est à 4 alors on le passe à 1
1436             self.page=1
1437             if self.page==1: #si page est à 1 on appelle la fonction page1()
1438                 self.page1()
1439             elif self.page==2: #si page est à 2 on appelle la fonction page2()
1440                 self.page2()
1441             elif self.page==3: #si page est à 3 on appelle la fonction page3()
1442                 self.page3()
1443
1444
1445     def page1(self): # cette fonction permet d'affiché seulement la page 1 des aeroport français
1446
1447         self.hide_page2() # on appelle la fonction hide_page2() qui va caché la page 2
1448
1449         self.hide_page3() # on appelle la fonction hide_page2() qui va caché la page 3
1450
1451         ##### on affiche tout Les aeroport suivant #####
1452         self.Hartsfield_Jackson.show()
1453         self.Pékin_Capitale.show()
1454         self.Dubaï.show()
1455         self.Los_Angeles.show()
1456         self.Tokyo_Haneda.show()
1457         self.Londres_Heathrow.show()
1458         self.Francfort_sur_le_Main.show()
1459         self.Paris_Charles_de_Gaulle.show()
1460         self.Atatürk.show()
1461         self.Paris_Orly.show()
1462         self.Lyon_Saint_Exupéry.show()
1463         self.Marseille_Provence.show()
1464         self.Toulouse_Blagnac.show()
1465         self.Bâle_Mulhouse_Fribourg.show()
1466         self.Bordeaux_Mérignac.show()
1467         self.Nantes_Atlantique.show()
1468         self.Beauvais_Tillé.show()
1469         self.Pointe_à_Pitre_Le_Raizet.show()
1470
1471     def page2(self): # cette fonction permet d'affiché seulement la page 1 des aeroport français
1472
1473         self.hide_page1() # on appelle la fonction hide_page1() qui va caché la page 1
1474
1475         ##### on affiche tout Les aeroport suivant #####

```

Ligne : 1433 – 1599

```

1471 def page2(self): # cette fonction permet d'affiché seulement la page 1 des aeroport français
1472
1473     self.hide_page1() # on appelle la fonction hide_page1() qui va caché la page 1
1474
1475     ##### on affiche tout les aeroport suivant #####
1476
1477     self.La_Réunion_Roland_Garros.show()
1478     self.Lille_Lesquin.show()
1479     self.Fort_de_France_Le_Lamentin.show()
1480     self.Montpellier_Méditerranée.show()
1481     self.Ajaccio_Napoléon_Bonaparte.show()
1482     self.Bastia_Poretta.show()
1483     self.Strasbourg_Entzheim.show()
1484     self.Brest_Bretagne.show()
1485     self.Biarritz_Pays_Basque.show()
1486     self.Rennes_Bretagne.show()
1487     self.Figari_Sud_Corse.show()
1488     self.Pau_Pyrénées.show()
1489     self.Nouméa_La_Tontouta.show()
1490     self.Cayenne_Félix_Eboué.show()
1491     self.Toulon_Hyères.show()
1492     self.Tarbes_Lourdes_Pyrénées.show()
1493     self.Nouméa_Magenta.show()
1494     self.Perpignan_Rivesaltes.show()
1495
1496
1497 def page3(self): # cette fonction permet d'affiché seulement la page 3 des aeroport français
1498
1499     self.hide_page2() # on appelle la fonction hide_page2() qui va caché la page 2
1500
1501     ##### on affiche tout les aeroport suivant #####
1502
1503     self.Clermont_Ferrand_Auvergne.show()
1504     self.Dzaoudzi_Pamandzi.show()
1505     self.Carcassonne_Salvaza.show()
1506     self.Calvi_Sainte_Catherine.show()
1507     self.Bora_Bora.show()
1508     self.Caen_Carpiquet.show()
1509     self.Lim.show()
1510     self.Bergerac_Dordogne_Périgord.show()
1511     self.Béziers_Cap_d'Agde.show()
1512     self.Metz_Nancy_Lorraine.show()
1513     self.Raïatea.show()
1514     self.La_Rochelle_Île_de_Ré.show()
1515     self.Nîmes_Alès_Camargue_Cévennes.show()
1516     self.Tours_Val_de_Loire.show()
1517     self.Lifou.show()
1518     self.Gustaf_III.show()

```

```

1517     self.Lifou.show()
1518     self.Gustaf_III.show()
1519     self.Huahine.show()
1520     self.Deauville_Normandie.show()
1521     self.Poitiers_Biard.show()
1522
1523
1524
1525 def hide_page1(self): # cette fonction permet de cahcé seulement la page 1 des aeroport français
1526
1527     ##### on cache tout les aeroports suivant #####
1528
1529     self.Hartsfield_Jackson.hide()
1530     self.Pékin_Capitale.hide()
1531     self.Dubai.hide()
1532     self.Los_Angeles.hide()
1533     self.Tokyo_Haneda.hide()
1534     self.Londres_Heathrow.hide()
1535     self.Francfort_sur_le_Main.hide()
1536     self.Paris_Charles_de_Gaulle.hide()
1537     self.Atatürk.hide()
1538     self.Paris_Orly.hide()
1539     self.Lyon_Saint_Exupéry.hide()
1540     self.Marseille_Provence.hide()
1541     self.Toulouse_Blagnac.hide()
1542     self.Bâle_Mulhouse_Fribourg.hide()
1543     self.Bordeaux_Mérignac.hide()
1544     self.Nantes_Atlantique.hide()
1545     self.Beauvais_Tillé.hide()
1546     self.Pointe_à_Pitre_Le_Raizet.hide()
1547
1548 def hide_page2(self): # cette fonction permet de cahcé seulement la page 2 des aeroport français
1549
1550     ##### on cache tout les aeroports suivant #####
1551
1552     self.La_Réunion_Roland_Garros.hide()
1553     self.Lille_Lesquin.hide()
1554     self.Fort_de_France_Le_Lamentin.hide()
1555     self.Montpellier_Méditerranée.hide()
1556     self.Ajaccio_Napoléon_Bonaparte.hide()
1557     self.Bastia_Poretta.hide()
1558     self.Strasbourg_Entzheim.hide()
1559     self.Brest_Bretagne.hide()
1560     self.Biarritz_Pays_Basque.hide()
1561     self.Rennes_Bretagne.hide()
1562     self.Figari_Sud_Corse.hide()
1563     self.Pau_Pyrénées.hide()

```

```

1561     self.Rennes_Bretagne.hide()
1562     self.Figari_Sud_Corse.hide()
1563     self.Pau_Pyrénées.hide()
1564     self.Nouméa_La_Tontouta.hide()
1565     self.Cayenne_Félix_Eboué.hide()
1566     self.Toulon_Hyères.hide()
1567     self.Tarbes_Lourdes_Pyrénées.hide()
1568     self.Nouméa_Magenta.hide()
1569     self.Perpignan_Rivesaltes.hide()
1570
1571     def hide_page3(self): # cette fonction permet de cacher seulement la page 3 des aeroport français
1572
1573         ##### on cache tout les aeroport suivant #####
1574
1575         self.Clermont_Ferrand_Auvergne.hide()
1576         self.Dzaoudzi_Pamandzi.hide()
1577         self.Carcassonne_Salvaza.hide()
1578         self.Calvi_Sainte_Catherine.hide()
1579         self.Bora_Bora.hide()
1580         self.Caen_Carpiquet.hide()
1581         self.Lim.hide()
1582         self.Bergerac_Dordogne_Périgord.hide()
1583         self.Béziers_Cap_d_Agde.hide()
1584         self.Metz_Nancy_Lorraine.hide()
1585         self.Raiatea.hide()
1586         self.La_Rochelle_Ile_de_Ré.hide()
1587         self.Nîmes_Alès_Camargue_Cévennes.hide()
1588         self.Tours_Val_de_Loire.hide()
1589         self.Lifou.hide()
1590         self.Gustaf_III.hide()
1591         self.Huahine.hide()
1592         self.Deauville_Normandie.hide()
1593         self.Poitiers_Biard.hide()
1594         self.Montluçon_Guéret.hide()
1595         self.Dole_Jura.hide()
1596         self.Lorient_Bretagne_Sud.hide()
1597         self.Montluçon_Guéret.hide()
1598         self.Dole_Jura.hide()
1599         self.Lorient_Bretagne_Sud.hide()

```

Change_page va nous permet de savoir dans quelle page on se trouve qui va nous permet de lancer la bonne fonction qui aura pour conséquence de cacher ou de montrer différents widgets en appelant 3 fonctions page1, page2, page3

Page1 va appeler 2 fonctions hide_page2, et hide_page 3 qui vont juste cacher tout ce qu'il se trouve dans ses 2 page en ce qui concerne le restant aéroport français on va les afficher on va utiliser le même principe pour page 2 et page 3 on va cacher la page précédent et afficher celle actuelle

Ligne : 1603 - 1636

```

1603     def changement_france(self): # cette fonction permet de cacher ou d'afficher les pages des aeroport français
1604         self.france_page=self.france_page+1
1605
1606         if self.france_page>=3: # si france_page est à 3 ou plus grand on la passe à 1
1607             self.france_page=1
1608
1609         if self.france_page==1: # si france_page est à 1 alors on affiche la page 1 des aeroport français
1610             self.suivant.show()
1611             self.toutselectionne.show()
1612             self.ligne6.show()
1613             self.page1()
1614
1615
1616         elif self.france_page==2: # si france_page est à 2 alors on cache toutes les pages des aeroport français
1617             self.suivant.hide()
1618             self.toutselectionne.hide()
1619             self.ligne6.hide()
1620
1621             self.hide_page1()
1622             self.hide_page2()
1623             self.hide_page3()
1624
1625
1626     def changement_autres(self): # cette fonction permet de cacher ou d'afficher les aeroport non français
1627         self.autre_page=self.autre_page+1
1628
1629         if self.autre_page>=3: # si autre_page est à 3 ou plus grand on la passe à 1
1630             self.autre_page=1
1631
1632         if self.autre_page==1: # si autre_page est à 1 alors on cache les aeroport non français
1633             self.autres_hide()
1634
1635         elif self.autre_page==2: # si autre_page est à 2 alors on affiche les aeroport non français
1636             self.autres_show()

```

Même principe que changement de page on regarde la valeur de la variable qui nous permettra de savoir si on doit cacher tous les aéroports de France ou afficher la page1 avec l'aide de la fonction page1 on utilise la même chose pour changements autres si on va afficher ou tout caché on va utiliser

deux fonctions autres_hide et autres_show

Ligne : 1639 – 1675

```
1639 def autres_hide(self): # cette fonctionne permet de caché Les aeroport non français
1640     self.ligne5.hide()
1641     self.aeroport1.hide()
1642     self.aeroport2.hide()
1643     self.aeroport3.hide()
1644     self.aeroport4.hide()
1645     self.aeroport5.hide()
1646     self.aeroport6.hide()
1647     self.aeroport7.hide()
1648     self.aeroport8.hide()
1649     self.aeroport9.hide()
1650     self.aeroport11.hide()
1651
1652 def autres_show(self): # cette fonctionne permet de d'affiché Les aeroport non français
1653     self.ligne5.show()
1654     self.aeroport1.show()
1655     self.aeroport2.show()
1656     self.aeroport3.show()
1657     self.aeroport4.show()
1658     self.aeroport5.show()
1659     self.aeroport6.show()
1660     self.aeroport7.show()
1661     self.aeroport8.show()
1662     self.aeroport9.show()
1663     self.aeroport11.show()
1664
1665 # main -----
1666 if __name__ == "__main__":
1667     print(' --- main --- ')
1668     # création d'une QApplication
1669     app = QApplication(sys.argv)
1670
1671     # creation d'un widget
1672     f = VueAnnuaire()
1673
1674     # Lancement de l'application
1675     sys.exit(app.exec())
```

Autres_hide va cacher les widget “ligne5” et les aéroports de 1 à 11, autres_show lui va faire l’inverse, puis il y a la main qui aura pour but d’appeler la fonction et de la lance le programme.

