```python
# -*- coding: utf-8 -*-
from pathlib import Path
import re

def getResults(file):
    """Prend en entrée un fichier de résultats et retourne une liste contenant les différents résultats"""
    data = file.read()
    data = re.sub(r"\n",r"",data)
    results = re.findall(r"<result>.*?</result>",data)
    return results

def getTag(result,tag):
    """Prend en entrée un résultat (un élément de la liste de résultats) et un nom de balise à rechercher et retourne le résultat pertinent contenu entre les balises souhaitées"""
    resu = re.findall(f"<{tag}>(.*?)</{tag}>",result)
    return resu

#Le script est commenté entièrement au début, puis seulement aux endroits où il y a des choses nouvelles car le principe reste toujours le même : on parse des fichiers de résultats en prenant en compte leur spécificité pour obtenir des fichiers contenant des Google arrays facilement copiables pour les insérer dans des graphiques.


#répertoire contenant les fichiers txt de résultats des requêtes XQuery
directory = Path("../requetes")
#répertoire de sortie qui contiendra les google arrays sous forme de fichiers txt
outDir = Path("../requetes_arrays")

#on parcourt le répertoire et on va faire des tests pour chaque sous-répertoire
for child in directory.iterdir():
    if child.is_dir():
        if child.match("*basic"):
            #on va parcourir tous les fichiers résultats du sous-répertoire "basic"
            for subchild in child.iterdir():
                #nom du fichier de sortie qui contiendra les résultats
                #sous forme de Google array
                nameOut = "basic/" + subchild.stem + "_array.txt"
                #on crée le chemin vers ce nouveau fichier
                pathToF = outDir.joinpath(nameOut)
                if subchild.match("*annee_r.txt"):
                    #on ouvre le premier fichier de résultats et on ouvre le fichier de sortie
                    with open(subchild,"r",encoding="utf-8") as file, open(pathToF, "a",encoding="utf8") as out:
                        #on récupère les résultats
                        results = getResults(file)
                        #on écrit les en-têtes
                        out.write('["Année", "Nombre de jeux"],\n')
                        #on parcourt les résultats pour récupérer toutes les informations
                        #que l'on écrit au fur et à mesure dans la sortie
                        #sous la forme souhaitée pour le graphique ensuite
                        for res in results:
                            year = getTag(res,"year")
                            count = getTag(res,"count")
                            row = f'["{year[0]}", {count[0]}],'
                            out.write(row)
                            out.write("\n")
                        print(f'Google array in "{nameOut}" generated ! \n')
                        print("------------------------------")

                if subchild.match("*decennies_r.txt"):
                    with open(subchild,"r",encoding="utf-8") as file, open(pathToF,"a",encoding="utf8") as out:
                        #ici on va faire attention de récupérer tous les genres pour les classer par ordre alphabétique
                        data = file.read()
                        data = re.sub(r"\n",r"",data)
                        listeGenre = getTag(data, "genre")
                        listeGenre = list(dict.fromkeys(listeGenre))
```

```
                        listeGenre.sort()
                        sep = '", "'
                        #on écrit l'en-tête avec les genres séparés par des virgules et entourés de guillemets
                        out.write(f'["Décennies", "{sep.join(listeGenre)}"],\n')
                        results = re.findall(r"<result>.*?</result>",data)
                        dicoR = {}
                        #on va récupérer les différentes informations des résultats
                        #et on va remplir un dictionnaire pour tout classer
                        for res in results:
                            year = getTag(res,"year")
                            count = getTag(res,"count")
                            genre = getTag(res,"genre")
                            if dicoR.get(genre[0]) is not None:
                               dicoR[genre[0]].append((year[0],str(count[0])))
                            else:
                               dicoR[genre[0]] = [(year[0],str(count[0]))]
                        liste1990 = []
                        liste2000 = []
                        liste2010 = []
                        #on va parcourir la liste des genres pour avoir le même ordre pour les trois décennies
                        #on connaît nos données donc on sait comment les gérer
                        #on gère les cas où il n'y a pas de données pour certains genres et certaines décennies
                        for genre in listeGenre:
                            result = dicoR.get(genre)
                            if len(result)==3:
                                liste1990.append(result[0][1])
                                liste2000.append(result[1][1])
                                liste2010.append(result[2][1])
                            elif len(result)==2:
                                liste1990.append("0")
                                liste2000.append(result[0][1])
                                liste2010.append(result[1][1])
                            else:
                                liste1990.append("0")
                                liste2000.append("0")
                                liste2010.append(result[0][1])
                        #on écrit tout dans la sortie avec les différentes listes de résultats par décennies dans le même ordre que l'en-tête
                        sep = ", "
                        row1 = f'["1990-1999", {sep.join(liste1990)}],'
                        out.write(row1)
                        out.write("\n")
                        row2 = f'["2000-2009", {sep.join(liste2000)}],'
                        out.write(row2)
                        out.write("\n")
                        row3 = f'["2010-2019", {sep.join(liste2010)}],'
                        out.write(row3)
                        out.write("\n")
                        print(f'Google array in "{nameOut}" generated ! \n')
                        print("------------------------------")

                    #les prochains traitements de fichiers résultats suivent la même logique que la précédente
                    if subchild.match("*genre_r.txt"):
                        with open(subchild,"r",encoding="utf-8") as file, open(pathToF, "a",encoding="utf8") as out:
                            results = getResults(file)
                            out.write('["Genre", "Nombre de jeux"],\n')
                            for res in results:
                                genre = getTag(res,"genre")
                                count = getTag(res,"count")
                                row = f'["{genre[0]}", {count[0]}],'
                                out.write(row)
                                out.write("\n")
                            print(f'Google array in "{nameOut}" generated ! \n')
                            print("------------------------------")

                    if subchild.match("*platform_r.txt"):
                        with open(subchild,"r",encoding="utf-8") as file, open(pathToF,
```

```
                    "a",encoding="utf8") as out:
128                     results = getResults(file)
129                     out.write('["Plateforme", "Nombre de jeux"],\n')
130                     for res in results:
131                         platform = getTag(res,"platform")
132                         count = getTag(res,"count")
133                         row = f'["{platform[0]}", {count[0]}],'
134                         out.write(row)
135                         out.write("\n")
136                     print(f'Google array in "{nameOut}" generated ! \n')
137                     print("------------------------------")
138 
139                 if subchild.match("*publisher_r.txt"):
140                     with open(subchild,"r",encoding="utf-8") as file, open(pathToF,
                    "a",encoding="utf8") as out:
141                         results = getResults(file)
142                         out.write('["Editeur", "Nombre de jeux"],\n')
143                         for res in results:
144                             publisher = getTag(res,"publisher")
145                             count = getTag(res,"count")
146                             row = f'["{publisher[0]}", {count[0]}],'
147                             out.write(row)
148                             out.write("\n")
149                         print(f'Google array in "{nameOut}" generated ! \n')
150                         print("------------------------------")
151 
152         if child.match("*plus_sales"):
153             for subchild in child.iterdir():
154                 nameOut = "plus_sales/" + subchild.stem + "_array.txt"
155                 pathToF = outDir.joinpath(nameOut)
156                 simFiles = ["genre_jeux_euro_r","genre_jeux_jap_r","genre_jeux_usa_r"]
157                 #certains fichiers résultats sont à gérer de la même manière. Pour
                    ne pas répéter le code on les liste
158                 if subchild.stem in simFiles:
159                     with open(subchild,"r",encoding="utf-8") as file, open(pathToF,
                    "a",encoding="utf8") as out:
160                         results = getResults(file)
161                         out.write('["Genre", "Ventes en millions"],\n')
162                         for res in results:
163                             genre = getTag(res,"genre")
164                             total = getTag(res,"total")
165                             row = f'["{genre[0]}", {total[0]}],'
166                             out.write(row)
167                             out.write("\n")
168                         print(f'Google array in "{nameOut}" generated ! \n')
169                         print("------------------------------")
170 
171                 simFiles2 = ["top10_ventes_euro_r","top10_ventes_global_r",
                    "top10_ventes_japon_r","top10_ventes_usa_r"]
172                 if subchild.stem in simFiles2:
173                     with open(subchild,"r",encoding="utf-8") as file, open(pathToF,
                    "a",encoding="utf8") as out:
174                         results = getResults(file)
175                         for res in results:
176                             name = getTag(res,"name")
177                             platform = getTag(res,"platform")
178                             sales = getTag(res,"sales")
179                             row = f'["{name[0]} ({platform[0]})", {sales[0]}],'
180                             out.write(row)
181                             out.write("\n")
182                         print(f'Google array in "{nameOut}" generated ! \n')
183                         print("------------------------------")
184 
185                 if subchild.match("*annees_global_r.txt"):
186                     with open(subchild,"r",encoding="utf-8") as file, open(pathToF,
                    "a",encoding="utf8") as out:
187                         results = getResults(file)
188                         out.write('["Année de sortie", "Ventes (en millions)"],\n')
189                         for res in results:
190                             year = getTag(res,"year")
191                             total = getTag(res,"total")
192                             row = f'["{year[0]}", {total[0]}],'
```

```
                            out.write(row)
                            out.write("\n")
                        print(f'Google array in "{nameOut}" generated ! \n')
                        print("------------------------------")

                if subchild.match("*ventes_zones_r.txt"):
                    with open(subchild,"r",encoding="utf-8") as file, open(pathToF,
                    "a",encoding="utf8") as out:
                        results = getResults(file)
                        out.write('["Pays", "Ventes (en millions)", "Continent"],\n')
                        for res in results:
                            total_usa = getTag(res,"total_sales_usa")
                            total_euro = getTag(res,"total_sales_eur")
                            total_jap = getTag(res,"total_sales_jap")
                        #on fait des listes de pays par continent car pour le
                        graphique sous forme de carte, on ne peut pas mélanger des
                        continents (Amérique du Nord et Europe) avec un pays
                        (Japon). Il est donc nécessaire de faire des listes de pays
                        afin de faire une carte propre où tout sera bien représenté
                        america = ["United States","Canada","Mexico"]
                        europe = ["France","Germany","Spain","Portugal","Italy",
                        "United Kingdom","Ireland","Poland","Russia","Ukraine",
                        "Sweden","Norway","Finland","Romania","Belarus","Kazakhstan",
                        "Greece","Bulgaria","Iceland","Hungary","Austria","Czech
                        Republic","Serbia","Lithuania","Latvia","Croatia","Bosnia
                        and Herzegovina","Slovakia","Estonia","Denmark","Switzerland"
                        ,"Netherlands","Moldova","Belgium","Armenia","Albania",
                        "North Macedonia","Turkey","Slovenia","Montenegro","Kosovo",
                        "Cyprus","Azerbaijan","Luxembourg","Georgia","Andorra",
                        "Malta","Liechtenstein","San Marino","Monaco","Vatican City"]
                        #pour chaque pays, on va écrire une ligne avec le bon total
                        et le bon continent
                        for country in america:
                            row = f'["{country}", {total_usa[0]}, "Amérique du
                            Nord"],'
                            out.write(row)
                            out.write("\n")

                        for country in europe:
                            row = f'["{country}", {total_euro[0]}, "Europe"],'
                            out.write(row)
                            out.write("\n")

                        row = f'["Japan", {total_jap[0]}, "Japon"],'
                        out.write(row)
                        out.write("\n")
                        print(f'Google array in "{nameOut}" generated ! \n')
                        print("------------------------------")

        if child.match("*plus_critic_sales"):
            for subchild in child.iterdir():
                nameOut = "plus_critic_sales/" + subchild.stem + "_array.txt"
                pathToF = outDir.joinpath(nameOut)
                if subchild.match("*genre_plat_r.txt"):
                    with open(subchild,"r",encoding="utf-8") as file, open(pathToF,
                    "a",encoding="utf8") as out:
                        results = getResults(file)
                        for res in results:
                            name = getTag(res,"name")
                            platform = getTag(res,"platform")
                            genre = getTag(res,"genre")
                            critic = getTag(res,"critic")
                            row = f'["{name[0]}", "{platform[0]}", "{genre[0]}",
                            {critic[0]}],'
                            out.write(row)
                            out.write("\n")
                        print(f'Google array in "{nameOut}" generated ! \n')
                        print("------------------------------")

                if subchild.match("*critic_sales_r.txt"):
                    with open(subchild,"r",encoding="utf-8") as file, open(pathToF,
                    "a",encoding="utf8") as out:
```

```
                        results = getResults(file)
                        out.write('["Titre du jeu", "Ventes mondiales", "Ventes
                        américaines", "Ventes européennes", "Ventes japonaises"],\n')
                        for res in results:
                            name = getTag(res,"name")
                            platform = getTag(res,"platform")
                            salesglob = getTag(res,"salesglob")
                            salesusa = getTag(res,"salesusa")
                            saleseuro = getTag(res,"saleseuro")
                            salesjap = getTag(res,"salesjap")

                            row = f'["{name[0]} ({platform[0]})", {salesglob[0]},
                            {salesusa[0]}, {saleseuro[0]}, {salesjap[0]}],'
                            out.write(row)
                            out.write("\n")
                        print(f'Google array in "{nameOut}" generated ! \n')
                        print("------------------------------")

        if child.match("*plus_esrb_sales"):
            for subchild in child.iterdir():
                nameOut = "plus_esrb_sales/" + subchild.stem + "_array.txt"
                pathToF = outDir.joinpath(nameOut)
                simFiles3 = ["esrb_ventes_euro_r","esrb_ventes_jap_r",
                "esrb_ventes_usa_r"]
                if subchild.stem in simFiles3:
                    with open(subchild,"r",encoding="utf-8") as file, open(pathToF,
                    "a",encoding="utf8") as out:
                        results = getResults(file)
                        out.write('["Symbole ESRB", "Ventes (en millions)"],\n')
                        for res in results:
                            esrb = getTag(res,"esrb")
                            sales = getTag(res,"sales")
                            row = f'["{esrb[0]}", {sales[0]}],'
                            out.write(row)
                            out.write("\n")
                        print(f'Google array in "{nameOut}" generated ! \n')
                        print("------------------------------")

                if subchild.match("*jeux_esrb_r.txt"):
                    with open(subchild,"r",encoding="utf-8") as file, open(pathToF,
                    "a",encoding="utf8") as out:
                        results = getResults(file)
                        out.write('["Symbole ESRB", "Nombre de jeux"],\n')
                        for res in results:
                            esrb = getTag(res,"esrb")
                            count = getTag(res,"count")
                            row = f'["{esrb[0]}", {count[0]}],'
                            out.write(row)
                            out.write("\n")
                        print(f'Google array in "{nameOut}" generated ! \n')
                        print("------------------------------")

                simFiles4 = ["top10_ventes_esrb_euro_r","top10_ventes_esrb_global_r",
                "top10_ventes_esrb_jap_r","top10_ventes_esrb_usa_r"]
                if subchild.stem in simFiles4:
                    with open(subchild,"r",encoding="utf-8") as file, open(pathToF,
                    "a",encoding="utf8") as out:
                        results = getResults(file)
                        for res in results:
                            name = getTag(res,"name")
                            platform = getTag(res,"platform")
                            esrb = getTag(res,"esrb")
                            sales = getTag(res,"sales")
                            row = f'["{name[0]} ({platform[0]})", "{esrb[0]}",
                            {sales[0]}],'
                            out.write(row)
                            out.write("\n")
                        print(f'Google array in "{nameOut}" generated ! \n')
                        print("------------------------------")
```