SAE 1.05 TRAITEMENT DE DONNEES

CE PROJET DE LA SAE 105 VISE À DÉVELOPPER UN **OUTIL DE REPORTING** POUR ANALYSER ET LOCALISER LES **GROS FICHIERS** PRÉSENTS SUR UN DISQUE DUR.

Objectifs :

- Libérer de **l'espace** en identifiant les fichiers volumineux.
- Fournir une solution **interactive** et **multiplateforme** (Windows, MacOS, Linux).
- Automatiser la **visualisation** et la **suppression** des fichiers grâce à des scripts Python et PowerShell.

- NOUMBA Kéran-lynch
- GARNIER Jonathan

LES DIFFÉRENTES PARTIES DES CODES

SAE 1.05

LES CODES SUR PYCHARM :

SCRIPT1

SEARCHGRAPH

SCRIPT3

SCRIPT1.PY

```python
def construire_liste_fichiers(repertoire_de_base):
    fichiers = []
    repertoire = Path(repertoire_de_base)
    for fichier in repertoire.rglob('*'):
        if fichier.is_file():
            infos_stat = fichier.stat()
            date_modif = datetime.datetime.fromtimestamp(infos_stat.st_mtime)
            fichiers.append({
                "Nom du fichier": fichier.name,
                "chemin_complet": str(fichier.resolve()),
                "taille_octets": infos_stat.st_size,
                "derniere_modification": date_modif.strftime('%Y-%m-%d %H:%M:%S')
            })
    return fichiers
```

Parcourt un répertoire et collecte les informations sur les fichiers :

Nom du fichier, chemin complet, taille en octet, dernière date de modification.

```python
def trier_par_taille_desc(liste_fichiers):
    # Trie la liste des fichiers par ordre décroissant de taille
    return sorted(liste_fichiers, key=lambda x: x["taille_octets"], reverse=True)
```

Trie les fichiers par ordre décroissant de taille.

```python
def filtrer_fichiers(liste_fichiers, taille_mini_mo, nb_max_fichiers):
    # Filtre les fichiers en fonction de la taille minimale (en Mo) et du nombre maximal spécifié
    taille_mini_octets = taille_mini_mo * 1024 * 1024
    fichiers_filtres = [f for f in liste_fichiers if f["taille_octets"] >= taille_mini_octets]
    return fichiers_filtres[:nb_max_fichiers]
```

Filtre les fichiers en fonction d'une taille minimale (en Mo) et d'un nombre maximal.

```python
def exporter_en_json(liste_fichiers, nom_fichier_json):
    # Crée un dossier pour le fichier JSON si nécessaire, puis l'exporte
    dossier_sortie = Path(nom_fichier_json).parent
    if not dossier_sortie.exists():
        dossier_sortie.mkdir(parents=True, exist_ok=True)

    for f in liste_fichiers:
        # Remplace les antislashs dans les chemins pour assurer la compatibilité avec Windows
        f["chemin_complet"] = f["chemin_complet"].replace('\\', '\\\\')

    with open(nom_fichier_json, 'w', encoding='utf-8') as fichier_json:
        json.dump(liste_fichiers, fichier_json, indent=4, ensure_ascii=False)
```

Exporte les données filtrées dans un fichier JSON, avec conversion correcte des chemins (antislashs).

```python
def main():
    # Crée l'application Qt
    app = QApplication(sys.argv)

    # Ouvre la boîte de dialogue de sélection de répertoire
    directory = QFileDialog.getExistingDirectory(None, "Choisir un répertoire")

    # Si l'utilisateur a sélectionné un dossier et validé
    if directory:
        print(directory)
    else:
        # En cas d'annulation, on renvoie une chaîne vide
        print("")

    sys.exit(0)

if __name__ == "__main__":
    main()
```

SEARCHG RAPH.PY

Fonction main :

- Utilisation de PyQt pour afficher une boîte de dialogue permettant à l'utilisateur de choisir un répertoire.
- Retourne le chemin du répertoire sélectionné ou une chaîne vide en cas d'annulation.

INTERFACE GRAPHIQUE ET CAMEMBERT (SCRIPT3.PY)

```python
def charger_liste_depuis_json(nom_fichier_json):
    chemin_json = Path(nom_fichier_json)
    if not chemin_json.exists():
        print(f"[ERREUR] Le fichier JSON '{nom_fichier_json}' est introuvable.")
        return []

    with open(chemin_json, 'r', encoding='utf-8') as f:
        data = json.load(f)

    liste_resultat = []
    if data and isinstance(data[0], list):
        # Format attendu : [ [chemin, taille], ... ]
        liste_resultat = data
    elif data and isinstance(data[0], dict):
        # Format alternatif : [ {"chemin_complet":..., "taille_octets":...}, ... ]
        for item in data:
            c = item.get("chemin_complet", "")
            t = item.get("taille_octets", 0)
            liste_resultat.append([c, t])
    else:
        liste_resultat = []
    return liste_resultat
```

Charge les données depuis le fichier JSON généré par le Scrip

```python
def generer_couleurs_aleatoires(nb_maxi_fichiers):
    couleurs = []
    for _ in range(nb_maxi_fichiers):
        r = random.randint(0, 255)
        g = random.randint(0, 255)
        b = random.randint(0, 255)
        couleurs.append(QColor(r, g, b))
    return couleurs
```

Génère des couleurs aléatoires pour les tranches du camembert.

```python
def callback_generer_script_ps1(self):
    if not self.selected_files:
        QMessageBox.warning(self, "Avertissement", "Aucun fichier sélectionné !")
        return

    default_name = str(Path(__file__).parent.joinpath("supprime_fichiers.ps1"))
    ps1_path, _ = QFileDialog.getSaveFileName(
        self,
        "Enregistrer le script PowerShell",
        default_name,
        "Fichier PowerShell (*.ps1)"
    )
    if not ps1_path:
        return  # Annulé
```

Crée et enregistre un script PowerShell pour supprimer les fichiers sélectionnés.

```python
def add_tabs_legendes(self):
    nb_par_onglet = 25
    for i in range(0, len(self.liste_fichiers), nb_par_onglet):
        subset = self.liste_fichiers[i : i + nb_par_onglet]

        tab = QWidget()
        layout = QVBoxLayout(tab)
        layout.setSpacing(8)

        for (chemin, taille) in subset:
            cb = QCheckBox(f"{chemin} ({taille} octets)")
            cb.stateChanged.connect(lambda state, f=chemin: self.toggle_file_selection(state, f))
            layout.addWidget(cb)

        self.tabs.addTab(tab, f"Légendes {int(i/nb_par_onglet) + 1}")

def toggle_file_selection(self, state, file_path):
    if state == Qt.Checked:
        self.selected_files.add(file_path)
    else:
        self.selected_files.discard(file_path)
```

Génère des onglets avec des cases à cocher pour les fichiers.

```python
def add_tab_camembert(self):
    tab = QWidget()
    layout = QVBoxLayout(tab)

    series = QPieSeries()
    total_size = sum(t for _, t in self.liste_fichiers)

    for i, (chemin, taille) in enumerate(self.liste_fichiers):
        label = Path(chemin).name
        slice_ = series.append(label, taille)

        # Affectation de la couleur de la tranche
        if i < len(self.couleurs):
            slice_.setColor(self.couleurs[i])

        # Calcul du pourcentage
        pct = (taille / total_size * 100) if total_size else 0

        # Affichage du label si le pourcentage est > 5%
        if pct > 5:
            slice_.setLabelVisible(True)
            slice_.setLabel(f"{label} ({pct:.1f}%)")

    chart = QChart()
    chart.addSeries(series)
    chart.setTitle("Répartition des fichiers par taille (Camembert)")

    # Police du titre en gras
    font_titre = QFont("Arial", 12, QFont.Bold)
    chart.setTitleFont(font_titre)

    chart.setAnimationOptions(QChart.SeriesAnimations)
    chart.legend().setFont(QFont("Arial", 10))

    chart_view = QChartView(chart)
    chart_view.setRenderHint(QPainter.Antialiasing)

    layout.addWidget(chart_view)
    self.tabs.addTab(tab, "Camembert")
```

RUN_SCRIPT.PS1

CLICK TO ADD FOOTER

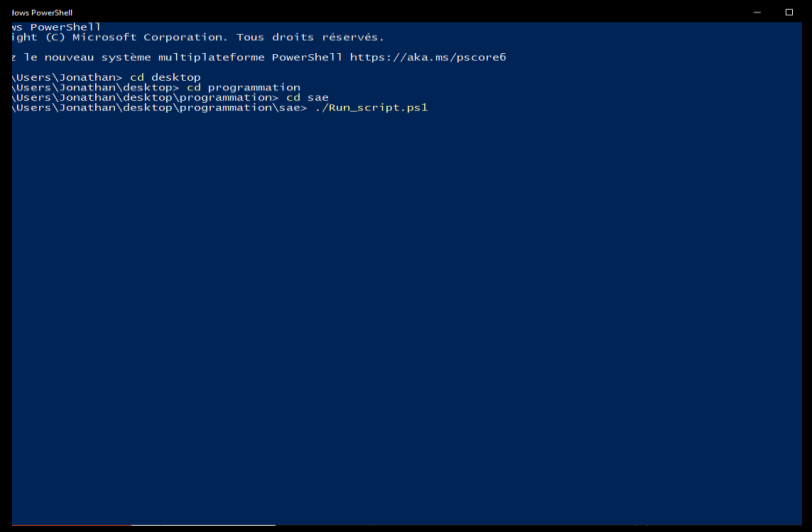
```powershell
# lancescripts.ps1
Param(
    [string]$pythonPath = "python"  # ou "C:\Python39\python.exe" si besoin
)

Write-Host " --> 1: Lancement de l'interface PyQt (searchgraph.py) pour sélectionner un répertoire <--"
```

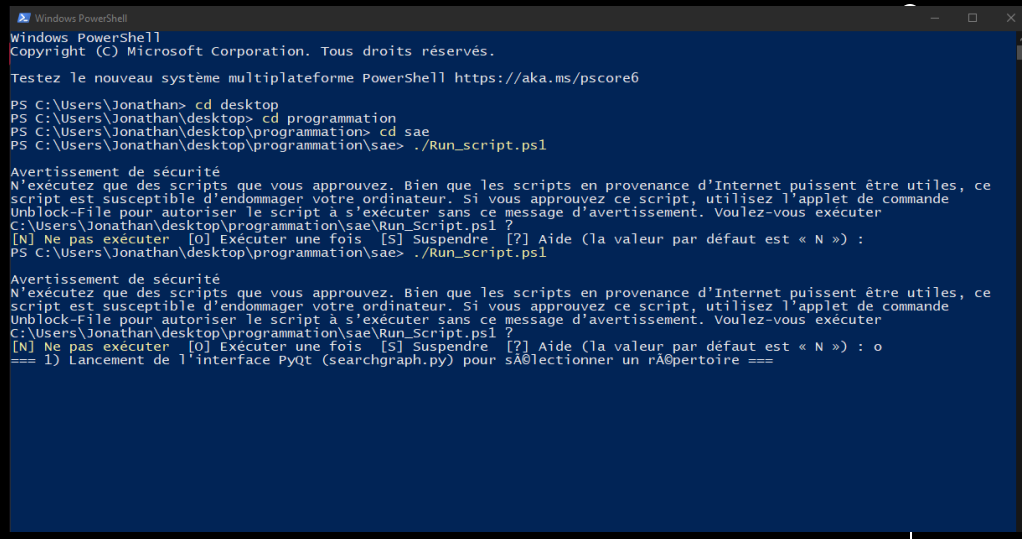
Lance le script1

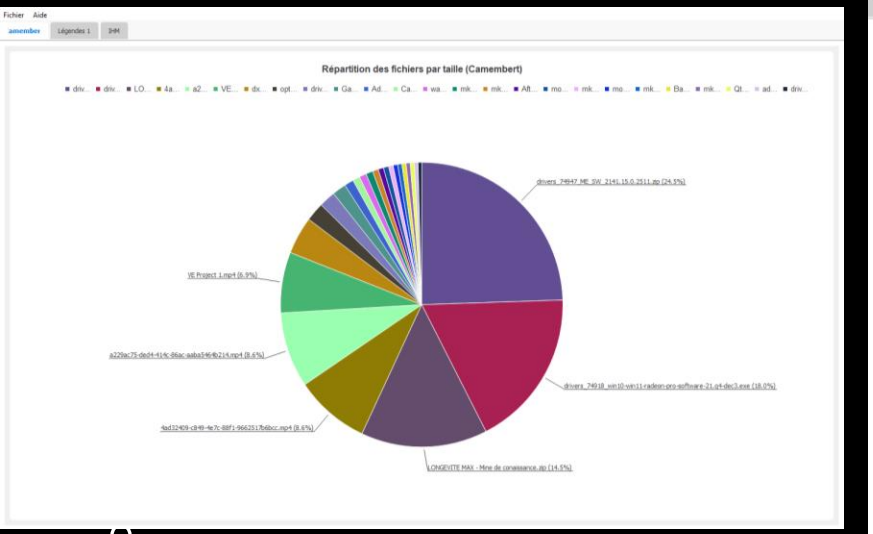
```powershell
# Lance searchgraph.py et récupère le répertoire sélectionné sur la sortie standard
$rep_base = & $pythonPath ".\searchgraph.py"

Write-Host "Répertoire sélectionné : '$rep_base'"
```

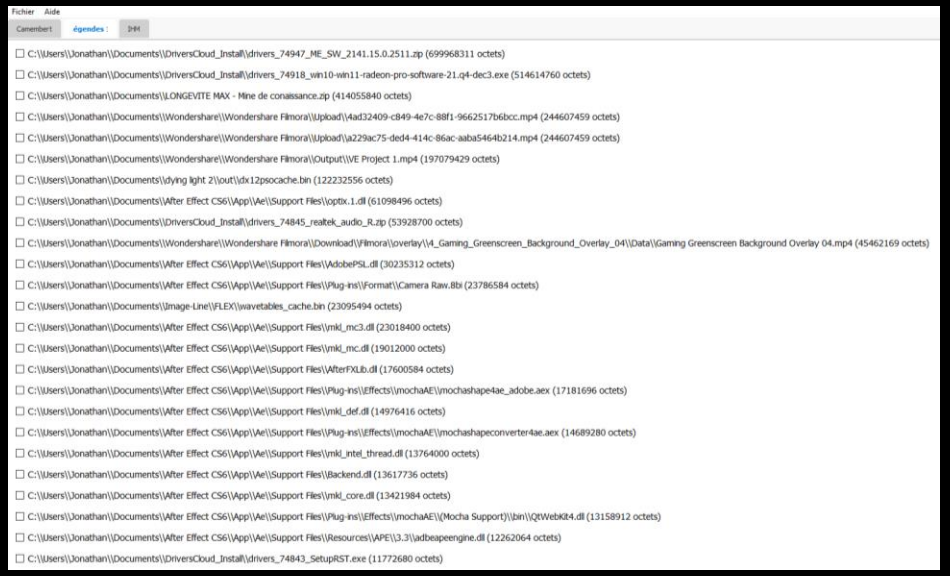
Lance le script searchgraph

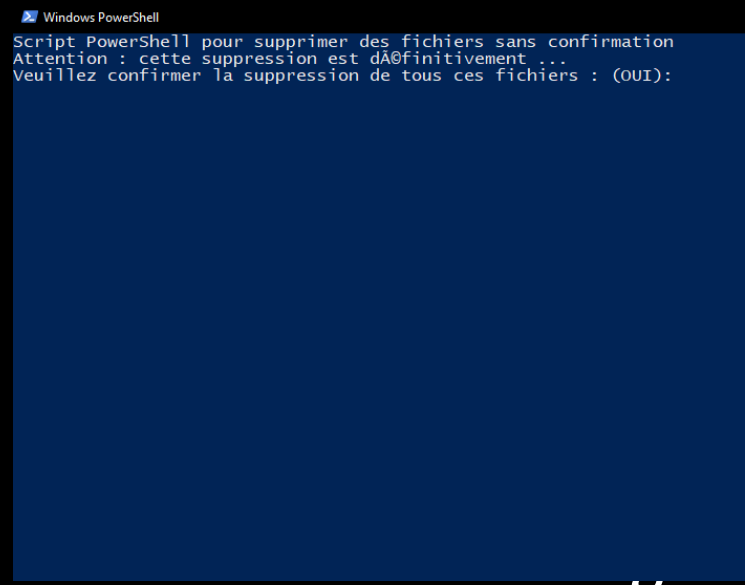
TEST :

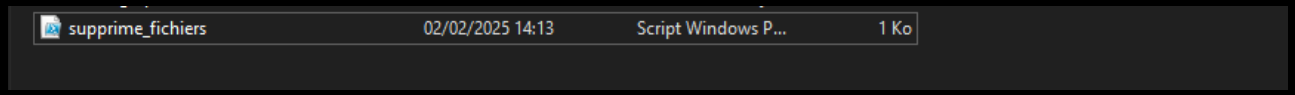

lows PowerShell
vs PowerShell
ight (C) Microsoft Corporation. Tous droits réservés.
z le nouveau système multiplateforme PowerShell https://aka.ms/pscore6
\Users\Jonathan> cd desktop
\Users\Jonathan\desktop> cd programmation
\Users\Jonathan\desktop\programmation> cd sae
\Users\Jonathan\desktop\programmation\sae> ./Run_script.ps1


Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.
Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6
PS C:\Users\Jonathan> cd desktop
PS C:\Users\Jonathan\desktop> cd programmation
PS C:\Users\Jonathan\desktop\programmation> cd sae
PS C:\Users\Jonathan\desktop\programmation\sae> ./Run_script.ps1
Avertissement de sécurité
N'exécutez que des scripts que vous approuvez. Bien que les scripts en provenance d'Internet puissent être utiles, ce script est susceptible d'endommager votre ordinateur. Si vous approuvez ce script, utilisez l'applet de commande Unblock-File pour autoriser le script à s'exécuter sans ce message d'avertissement. Voulez-vous exécuter C:\Users\Jonathan\desktop\programmation\sae\Run_Script.ps1 ?
[N] Ne pas exécuter [O] Exécuter une fois [S] Suspendre [?] Aide (la valeur par défaut est « N ») :
PS C:\Users\Jonathan\desktop\programmation\sae> ./Run_script.ps1
Avertissement de sécurité
N'exécutez que des scripts que vous approuvez. Bien que les scripts en provenance d'Internet puissent être utiles, ce script est susceptible d'endommager votre ordinateur. Si vous approuvez ce script, utilisez l'applet de commande Unblock-File pour autoriser le script à s'exécuter sans ce message d'avertissement. Voulez-vous exécuter C:\Users\Jonathan\desktop\programmation\sae\Run_Script.ps1 ?
[N] Ne pas exécuter [O] Exécuter une fois [S] Suspendre [?] Aide (la valeur par défaut est « N ») : o
=== 1) Lancement de l'interface PyQt (searchgraph.py) pour sA©lectionner un rA©pertoire ===


Fichier Aide
Répartition des fichiers par taille (Camembert)


Fichier Aide
C:\\Users\\Jonathan\\Documents\\DriversCloud_Install\\drivers_74947_ME_SW_2141.15.0.2511.zip (699968311 octets)
C:\\Users\\Jonathan\\Documents\\DriversCloud_Install\\drivers_74918_win10-win11-radeon-pro-software-21.q4-dec3.exe (514614760 octets)
C:\\Users\\Jonathan\\Documents\\LONGEVITE MAX - Mine de conaissance.zip (414055840 octets)
C:\\Users\\Jonathan\\Documents\\Wondershare\\Wondershare Filmora\\Upload\\4ad32409-c849-4e7c-88f1-9662517b6bcc.mp4 (244607459 octets)
C:\\Users\\Jonathan\\Documents\\Wondershare\\Wondershare Filmora\\Upload\\a229ac75-ded4-414c-86ac-aaba5464b214.mp4 (244607459 octets)
C:\\Users\\Jonathan\\Documents\\Wondershare\\Wondershare Filmora\\Output\\VE Project 1.mp4 (197079429 octets)
C:\\Users\\Jonathan\\Documents\\dying light 2\\out\\dx12psocache.bin (122232556 octets)
C:\\Users\\Jonathan\\Documents\\After Effect CS6\\App\\Ae\\Support Files\\optix.1.dll (61098496 octets)
C:\\Users\\Jonathan\\Documents\\DriversCloud_Install\\drivers_74845_realtek_audio_R.zip (53928700 octets)
C:\\Users\\Jonathan\\Documents\\Wondershare\\Wondershare Filmora\\Download\\Filmora\\overlay\\4_Gaming_Greenscreen_Background_Overlay_04\\Data\\Gaming Greenscreen Background Overlay 04.mp4 (45462169 octets)
C:\\Users\\Jonathan\\Documents\\After Effect CS6\\App\\Ae\\Support Files\\AdobePSL.dll (30235312 octets)
C:\\Users\\Jonathan\\Documents\\After Effect CS6\\App\\Ae\\Support Files\\Plug-ins\\Format\\Camera Raw.8bi (23786584 octets)
C:\\Users\\Jonathan\\Documents\\Image-Line\\FLEX\\wavetables_cache.bin (23095494 octets)
C:\\Users\\Jonathan\\Documents\\After Effect CS6\\App\\Ae\\Support Files\\mkl_mc3.dll (23018400 octets)
C:\\Users\\Jonathan\\Documents\\After Effect CS6\\App\\Ae\\Support Files\\mkl_mc.dll (19012000 octets)
C:\\Users\\Jonathan\\Documents\\After Effect CS6\\App\\Ae\\Support Files\\AfterFXLib.dll (17600584 octets)
C:\\Users\\Jonathan\\Documents\\After Effect CS6\\App\\Ae\\Support Files\\Plug-ins\\Effects\\mochaAE\\mochashape4ae_adobe.aex (17181696 octets)
C:\\Users\\Jonathan\\Documents\\After Effect CS6\\App\\Ae\\Support Files\\mkl_def.dll (14976416 octets)
C:\\Users\\Jonathan\\Documents\\After Effect CS6\\App\\Ae\\Support Files\\Plug-ins\\Effects\\mochaAE\\mochashapeconverter4ae.aex (14689280 octets)
C:\\Users\\Jonathan\\Documents\\After Effect CS6\\App\\Ae\\Support Files\\mkl_intel_thread.dll (13764000 octets)
C:\\Users\\Jonathan\\Documents\\After Effect CS6\\App\\Ae\\Support Files\\Backend.dll (13617736 octets)
C:\\Users\\Jonathan\\Documents\\After Effect CS6\\App\\Ae\\Support Files\\mkl_core.dll (13421984 octets)
C:\\Users\\Jonathan\\Documents\\After Effect CS6\\App\\Ae\\Support Files\\Plug-ins\\Effects\\mochaAE\\(Mocha Support)\\bin\\QtWebKit4.dll (13158912 octets)
C:\\Users\\Jonathan\\Documents\\After Effect CS6\\App\\Ae\\Support Files\\Resources\\APE\\3.3\\adbeapeengine.dll (12262064 octets)
C:\\Users\\Jonathan\\Documents\\DriversCloud_Install\\drivers_74843_SetupRST.exe (11772680 octets)


Windows PowerShell
Script PowerShell pour supprimer des fichiers sans confirmation
Attention : cette suppression est dA©finitivement ...
Veuillez confirmer la suppression de tous ces fichiers : (OUI):


supprime_fichiers
02/02/2025 14:13
Script Windows P...
1 Ko

AVANTAGES ET SÉCURITÉ

- Gain de temps
- Visualisation intuitive
- Interaction simplifiée
- Automatisation

- Contrôle utilisateur
- Prévention des erreurs
- Protection des fichiers critiques

RÉSULTATS

Le projet a permis de mettre en place une solution fonctionnelle répondant aux objectifs. Voici les principaux résultats observés :

- Le script a détecté, trié et filtré les fichiers volumineux d'un répertoire de test.
- Les résultats sont exportés dans un fichier JSON structuré.
- La visualisation graphique affiche clairement les fichiers les plus volumineux.
- La sélection de fichiers suivie de la génération du script PowerShell s'est faite avec succès, démontrant l'efficacité de l'automatisation du processus de suppression.

CONCLUSION

Perspectives d'amélioration :

- Ajout d'autres formats de visualisation, comme des graphiques en barres.
- Intégration de notifications automatiques en cas de dépassement d'un seuil critique d'espace disque.
- Possibilité de configurer des filtres
  - avancés selon le type
  - ou l'âge des fichiers.

https://github.com/R31NR4G/SAE

Bénéfices :

- Optimisation de la gestion de l'espace disque.
- Simplification de l'analyse et de la suppression des fichiers volumineux.
- Solution personnalisée adaptée aux besoins réels des utilisateurs.