gui\_create\_txtFileForDownload.py

# **1** **Giới thiệu**

Chương trình được phân tích là một ứng dụng giao diện người dùng (GUI) được viết bằng Python, sử dụng thư viện tkinter. Mục đích chính của chương trình là nhập dữ liệu từ người dùng (một file có format json là api được trả về sau khi nhấn vào xem một hscene), lưu dữ liệu vào một file văn bản với tên dựa trên số thứ tự (STT) đọc từ file stt.txt, và sau đó gọi hàm xử lý file từ module downloadfile. Chương trình có thể được sử dụng để quản lý và xử lý danh sách các tài nguyên tải xuống (ví dụ: file video .mp4 hoặc các file khác).

# **2** **Mục đích của chương trình**

• **Nhậpdữliệu**: Người dùng nhập hoặc dán danh sách dữ liệu vào một vùng văn bản (ScrolledText) bằng cách nhấn nút paste, dữ liệu này là nội dung văn bản có định dạng json hoặc list trong python , văn bản này là kết quả trả về khi nhấn vào xem một h - scene.

Lưu ý: khi gặp một nhân vật có cảnh H động, thì phải chọn xem cảnh động trước rồi mới xem cảnh thường

• **Lưudữliệu**:Dữ liệu được lưu vào file văn bản trong thư mục C:\Users\Admin\Downloa resource\ với tên file dựa trên giá trị STT.

• **Xửlýdữliệu**: File được truyền vào hàm downloadfile.process\_file() để tải file. Tham số truyền vào sẽ là:

* Stt số thứ tự của h - scene, là một String
* địa chỉ của file vừa được lưu
* flag animated- True nếu có cảnh h scene động, False nếu không có

• **Quảnlýsốthứtự(STT)**: Nếu dữ liệu không chứa file .mp4 nghĩa là cảnh h động, số thứ tự trong stt.txt sẽ tăng lên.

• **Đóng ứng dụng**: Ứng dụng tự động đóng sau khi lưu và xử lý thành công.

# **3** **Input**

• **Dữ liệu người dùng nhập**:

– Là một chuỗi dài,được copy thẳng từ api khi xem một h scene.

– định dạng json, chứa link nội dung kịch bản (scene scenario), file âm thanh (.mp3) chỉ chứa một trong hai

* hoặc là file hình ảnh .jpg
* hoặc là file video .mp4

– Người dùng có thể dán dữ liệu từ clipboard bằng nút ”Paste from Clipboard”.

• **File stt.txt**:

– Đọc từ C:\Users\Admin\Downloads\stt.txt.

– Nội dung là một số nguyên (số thứ tự) để xác định tên file lưu dữ liệu (ví dụ: 1.txt, 2.txt, hoặc 1a.txt nếu có file .mp4).

**Lưu ý về input**:

• Dữ liệu nhập không được để trống, nếu không sẽ hiển thị cảnh báo.

• File stt.txt phải tồn tại và chứa số hợp lệ, nếu không chương trình sẽ báo lỗi.

# **4** **Output**

• **File văn bản**:

– Lưu trong thư mục C:\Users\Admin\Downloads\d1 resource\. **–** Tên file:

\* Nếu **không chứa file .mp4**: {stt}.txt (ví dụ: 1.txt, 2.txt).

\* Nếu **chứa file .mp4**: {stt}a.txt (ví dụ: 1a.txt, 2a.txt).

– Nội dung file là các dòng dữ liệu nhập, mỗi dòng kết thúc bằng ký tự xuống dòng (\n).

\_ Mục đích là lưu lại cái ta đã copy từ hscene

• **Cập nhật stt.txt**:

– Nếu dữ liệu không chứa .mp4, giá trị trong stt.txt tăng lên 1 (ví dụ: từ 1 thành 2).

– Nếu chứa .mp4, stt.txt không thay đổi.

• **Xử lý file**: File được truyền vào hàm downloadfile.process\_file() để xử lý tiếp.

• **Thôngbáo**:Hiểnthịthôngbáothànhcônghoặclỗiquahộpthoại(messagebox).

# **5** **Format và lưu ý**

• **Format của input**:

– Danh sách chuỗi, mỗi chuỗi trên một dòng.

– Chương trình kiểm tra sự hiện diện của .mp4 để quyết định tên file và cách xử lý.

• **Format của output**:

– File văn bản: Mỗi dòng là một chuỗi từ input, kết thúc bằng \n.

– Tên file: {stt}.txt hoặc {stt}a.txt.

• **Lưu ý quan trọng**:

1. **Đườngdẫncốđịnh**:Chươngtrìnhsửdụngđườngdẫncứng(C:\Users\Admin\Down chỉ hoạt động trên máy tính có thư mục và tài khoản phù hợp.

2. **Moduledownloadfile**:Phụthuộcvàomoduledownloadfilevàhàm process\_file, cần đảm bảo module này tồn tại.

3. **Xửlýlỗi**: Có xử lý lỗi cơ bản, nhưng không kiểm tra sự tồn tại của thư mục d1 resource.

4. **Tính năng clipboard**: Nút ”Paste from Clipboard” cho phép dán dữ liệu nhanh, nhưng báo lỗi nếu clipboard rỗng.

5. **Tự động đóng**: Ứng dụng đóng sau khi lưu thành công.

6. **Hạn chế về định dạng**: Không kiểm tra định dạng dữ liệu đầu vào kỹ lưỡng, có thể gây lỗi nếu module downloadfile yêu cầu định dạng đặc biệt.

# **6** **Tóm tắt**

• **Mục đích**: Nhập danh sách dữ liệu, lưu vào file văn bản với tên dựa trên STT, và xử lý file bằng module downloadfile.

• **Input**: Danh sách chuỗi (nhập tay hoặc từ clipboard), file stt.txt chứa số thứ tự.

• **Output**: File văn bản {stt}.txt hoặc {stt}a.txt, cập nhật stt.txt (nếu không có .mp4), và gọi hàm xử lý file.

• **Lưuý**: Đường dẫn cứng, phụ thuộc module downloadfile, không kiểm tra định dạng dữ liệu kỹ lưỡng, ứng dụng tự đóng sau khi lưu.

Phân tích file downloadfile.py

# **1** **Giới thiệu**

File downloadfile.py là một module Python hỗ trợ xử lý việc tải xuống các tài nguyên (như file MP3, JPG, MP4) từ một URL cụ thể bằng cách sử dụng **Internet Download Manager (IDM)**. Module này liên kết chặt chẽ với chương trình GUI (sử dụng tkinter) đã phân tích trước đó, cung cấp hàm process\_file để xử lý file JSON chứa thông tin tài nguyên, tạo URL tải xuống, và thêm vào hàng đợi IDM. Mục đích chính là tự động hóa việc tải các tài nguyên về máy theo cấu trúc thư mục và tên file được quy định.

# **2** **Mục đích của chương trình**

• **Xử lý file JSON**: Đọc file JSON chứa danh sách tài nguyên (URL, tên file, mã MD5).

• **Tạo URL tải xuống**: Tạo URL đầy đủ từ thông tin trong JSON và URL cơ sở.

• **ThêmvàohàngđợiIDM**: Gọi lệnh hệ thống để thêm URL vào hàng đợi IDM với đường dẫn lưu và tên file phù hợp.

• **Quản lý đường dẫn lưu**: Quy định thư mục và tên file dựa trên loại file (MP3, JPG, MP4) và số thứ tự (stt).

• **Kiểm tra file tồn tại**: Bỏ qua file đã tồn tại để tránh tải lại.

Chương trình kết hợp với GUI để tạo hệ thống: GUI nhập danh sách tài nguyên, lưu thành file JSON, và downloadfile.py xử lý file này để tải về thông qua IDM.

# **3** **Input**

## **3.1** **Tham số của hàm process\_file**

Hàm process\_file(stt, filepath, animated=False) nhận:

• stt: Số thứ tự (string, ví dụ: ”1051”) để xác định tên file và thư mục lưu.

• filepath:ĐườngdẫnđếnfileJSON(vídụ:C:\Users\Admin\Downloads\d1 resource\1051.txt).

• animated: Boolean, xác định danh sách có chứa file MP4 hay không (True nếu có, False nếu không).

## **3.2** **File JSON**

• **Đường dẫn**: C:\Users\Admin\Downloads\d1 resource\.

• **Định dạng**: Chứa khóa ”resource” là danh sách các dictionary, mỗi dictionary có:

– ”path”: Đường dẫn tương đối trên server.

– ”md5”: Mã MD5 của file.

– ”fileName”: Tên file với phần mở rộng (ví dụ: hvoice\_1051.mp3).

• **Ví dụ**:

{

”resource”: [

{”path”: ”audio/123”, ”md5”: ”abc123”, ”fileName”: ”hvoic

{”path”: ”video/456”, ”md5”: ”xyz789”, ”fileName”: ”ani 1

]

}

## **3.3** **Cấu hình cố định**

• IDM\_PATH:C:\Program Files (x86)\Internet Download Manager\IDMan.exe.

• BASE\_URL:https://tonofura-r-cdn-resource.deepone-online.com/deep\_one/d

• BASE\_SAVE\_PATH:C:\Users\Admin\Downloads\d1 resource downloaded. **Lưu ý về input**:

• File JSON phải tồn tại, đúng định dạng, nếu không chương trình dừng.

• Chỉ xử lý file .txt, .mp3, .jpg, .mp4; các loại khác bị bỏ qua.

• Nếu animated=False, file MP4 không được xử lý.

# **4** **Output**

## **4.1** **Hàng đợi IDM**

• URL tải được thêm vào IDM bằng lệnh:

IDMan.exe /a /d <url> /p <save\_folder> /f <save\_name> /n

• Người dùng cần mở IDM và chọn ”Start Queue” để tải.

## **4.2** **Thư mục và file lưu**

• tùy loại file:**Thưmục**:TrongC:\Users\Admin\Downloads\d1 resource downloaded,

– .txt: <BASE\_SAVE\_PATH>\<stt>\{stt}.txt hoặc {stt}a.txt nếu animated=True.

– .mp3:<BASE\_SAVE\_PATH>\<stt>\hvoice\_{stt}\hvoice\_{stt}.mp3.

– .jpg: <BASE\_SAVE\_PATH>\<stt>\hcg\_{stt}\hcg\_hcg\_{stt}.jpg.

– .mp4:<BASE\_SAVE\_PATH>\ani\ani\_{stt}\ani {stt}.mp4nếuanimated=True.

• Thư mục được tạo tự động nếu chưa tồn tại.

• File đã tồn tại sẽ bị bỏ qua.

## **4.3** **Thông báo console**

• Đường dẫn file lưu.

• Bỏ qua (file đã tồn tại): <save\_path>.

• Đưa vào queue: <download\_url> -> <save\_path>.

• Đã xử lý xong {stt}. Mở IDM -> Start Queue để tải nhé..

**Lưu ý về output**:

• Không tạo file trực tiếp, chỉ thêm vào hàng đợi IDM.

• File thực tế xuất hiện sau khi IDM tải xong.

• Tên và thư mục lưu được quy định chặt chẽ.

# **5** **Format và lưu ý**

## **5.1** **Format của input**

• **File JSON**: Khóa ”resource” chứa danh sách dictionary với path, md5, fileName.

• **Tham số**: stt là chuỗi, filepath trỏ đến JSON, animated là boolean.

## **5.2** **Format của output**

• **URL tải**: <BASE\_URL>/<file\_path>/<file\_md5><ext>.

• **Tên file lưu**: {stt}.txt, hvoice\_{stt}.mp3, hcg\_hcg\_{stt}.jpg, ani {stt}.mp4.

• **Thư mục**: Cấu trúc rõ ràng theo loại file.

## **5.3** **Lưu ý quan trọng**

1. **PhụthuộcIDM**: Yêu cầu IDM cài tại C:\Program Files (x86)\.... Lệnh subprocess.call có thể thất bại nếu IDM không tồn tại.

2. **Đường dẫn cố định**: Chỉ hoạt động trên máy có tài khoản Admin và thư mục phù hợp.

3. **Định dạng JSON**: Phải đúng, nếu không gây lỗi khi đọc.

4. **Xử lý MP4**: Chỉ xử lý MP4 nếu animated=True.

5. **Kiểm tra file**: Chỉ kiểm tra sự tồn tại, không xác minh nội dung.

6. **Hiệu suất**: Danh sách dài có thể chậm khi thêm vào IDM.

7. **Xử lý lỗi**: Chưa đầy đủ (ví dụ: không kiểm tra lỗi IDM, quyền ghi thư mục).

8. **Liên kết với GUI**: File JSON phải khớp với output của GUI; cần nhất quán về stt và animated.

# **6** **Mối liên hệ với chương trình GUI**

• **GUI**:Nhậpdanhsáchtàinguyên,lưuthànhfile.txthoặcJSON,gọiprocess\_file(stt, filepath, animated).

• **downloadfile.py**: Đọc JSON, tạo URL, thêm vào IDM.

• **Luồng hoạt động**:

1. Người dùng nhập danh sách vào GUI.

2. GUI lưu thành file .txt hoặc JSON (có thể có bước chuyển đổi .txt sang JSON).

3. GUI gọi process\_file để xử lý.

4. Người dùng mở IDM để tải.

• **Lưu ý**: Có thể cần bước chuyển đổi .txt sang JSON (chưa được cung cấp).

# **7** **Tóm tắt**

• **Mục đích**: Xử lý file JSON, tạo URL tải, thêm vào hàng đợi IDM để tải MP3, JPG, MP4.

• **Input**: Tham số stt, filepath, animated; file JSON với khóa resource.

• **Output**: Hàng đợi IDM, file lưu theo cấu trúc, thông báo console.

• **Format**: JSON với path, md5, fileName; file lưu theo quy tắc tên/thư mục.

• **Lưu ý**: Phụ thuộc IDM, đường dẫn cứng, JSON đúng định dạng, xử lý lỗi chưa đầy đủ, cần nhất quán với GUI.