**Project Status Report I: Dash Cam Video Joiner**

**Name of the Application**:   
  
Dash Cam Video Joiner

**Purpose of the Application:**

The purpose of Dash Cam Video Joiner is to provide a convenient and automated way to merge multiple dash cam video files based on their timestamps. Dash cam videos are typically broken up into short minute intervals for several data safety reasons. The program will monitor a specified directory for new video files (A webdav or network share preferably) and seamlessly join them together as video files are synced over the network, allowing users to create longer, contiguous video segments without manual intervention. This tool is particularly useful for users who need to compile video footage from dash cams for extended periods, such as for road trips, security, hobbyist, or legal purposes.

**Reason for Creating the Application:**

I’m creating this application to streamline the process of managing and merging dash cam videos. Dash cams typically record in short, segmented clips, and manually joining these files can be tedious and error-prone. By automating this process, the application will save time and reduce the risk of human error. Additionally, this project allows me to apply my knowledge of Python, video processing, and GUI development while providing a practical solution for a common problem. I am also a hobbyist dash-cammer, and I’ve really wanted to make something like this for a long time.

**Goals for the Application:**

- Automate the process of monitoring a directory for new video files.

- Seamlessly join video files based on their timestamps, ensuring chronological order.

- Provide a user-friendly interface where users can set time thresholds and control the monitoring process.

- Ensure robust error handling and logging to track operations and handle potential issues with video files.

- Efficiently manage large numbers of video files and optimize performance for long-term use.

**Target Audience:**

The target audience for this application includes a wide range of users who rely on dash cams for personal, security, or work purposes. This could include individuals who record their road trips, work drives, vacations, security personnel who monitor traffic or public areas, or legal professionals who need to compile video evidence. The audience is likely to be adults aged 16-60, with moderate to high levels of technical comfort. Several audiences may find inherent utility in something like this.

**Outline of the Final Python tkinter GUI Application**:

**1. Main Window (Tkinter GUI)**:

- A simple, intuitive interface that allows users to:

- Start and stop the monitoring process.

- Display the status and log of the joining process, including progress and errors.  
 - Detect new video files as they are added to the directory and process them based on their timestamps.  
 - Hit an additional button to configure the applications settings and directory.  
 - Exit the Application.

**3. Settings and Directory Monitoring Page:**

- The application will monitor a directory using the `watchdog` library.  
 - The user will select the directory to monitor for video files.  
 - The user can input a time threshold (e.g., 90 seconds) to determine which videos should be joined. (So the program knows how long to look for between videos depending on your model of camera and how long it cuts/segments videos.)

**3. Time Analysis and Video Joining:**

- The application will use the `MoviePy` and `FFmpeg` libraries to handle video joining.

- It will parse video filenames to extract timestamp information and calculate the time difference between files.

- Videos within the specified time threshold will be joined together in chronological order.

**4. Error Handling and Logging:**

- The application will include robust error handling to manage issues such as file access errors, corrupt videos, or unexpected filename formats.

- A log will be maintained to track all operations and errors.

**5. Performance Optimization:**

- The application will be optimized to handle large volumes of video files.

- Original video files will be deleted after verified to be successfully joined to conserve disk space.

This project will provide a practical, user-friendly solution for managing dash cam video files, while also allowing me to demonstrate my skills in Python, video processing, and GUI development.

Github: <https://github.com/PaulSommers/SDEV140-Project-Status-Report1>