MTRE 6100, Assignment 5

Enhanced Video Processing with Object-Oriented Programming

Part A: Class Design for Video Handling:

- **Task**: Develop a program focused on video processing using Object-Oriented Programming principles.
- **Language**: You may use either Python or C++.
- **Base Class**: Create a base class named `videoProcessor` that handles video loading, based on the name and path provided by the user.

    - **Functions**:
        - `load_video()`: Load a video from a specified path and read its frames for processing.
        - `display_info()`: Print out basic video information such as resolution, frame rate, and duration.
        - `play_video()`: Play the video using OpenCV's video display functionality.
        - `rewind_video()`: Implement a function that allows the user to rewind the video. This can be a full rewind to the beginning or a more controlled rewind (e.g., 10 seconds back), depending on user input.

Part B: Advanced Video Processing:

- **Derived Class**: Add a new class named `objectHighlighter` that **inherits** from `videoProcessor`.
    - **Methods**:
        - `object_selection()`: Play the video to a specific frame, pause, and allow the user to select objects. The selection mechanism is up to you, but the user must be able to select multiple objects, either one at a time or several at once.
        - `highlight_objects()`: Override the `play_video()` function to play the video with the selected objects highlighted using an opaque overlay in a different color (e.g., semi-transparent blue or green).
        - `save_video_with_highlights()`: Save the video with the highlighted objects, including an option to save in different formats (e.g., MP4, AVI).
        - `capture_frame_with_highlights()`: Save a specific frame with the highlighted objects.

Extra Features:

- Implement advanced selection methods like freehand or point-based selection (optional and extra credit).
- Add functionality to track the selected objects across frames and update the highlights dynamically.
- Include an option to apply additional visual effects to the highlighted objects (e.g., blur or color change).

Delivery:

1. A PDF with a screenshot showing the successful object highlighting in the video with the opaque overlay.
2. Your source code and any relevant material (e.g., CMakeLists.txt if using C++).
3. Ensure best practices in coding:
    - Comment your code and provide a clear explanation of your logic.
    - Organize your code with reusable functions/methods.
    - Maintain consistent naming conventions and clean up unused code.