

Product Name: RePoste

Client: Terry Yoo

Team Name: Goobernauts

Team Member Names:

Stephen Goodridge, Logan Geiser, David Geng, Ryan Giles,
Heath Miller

Date: 11/12/2024

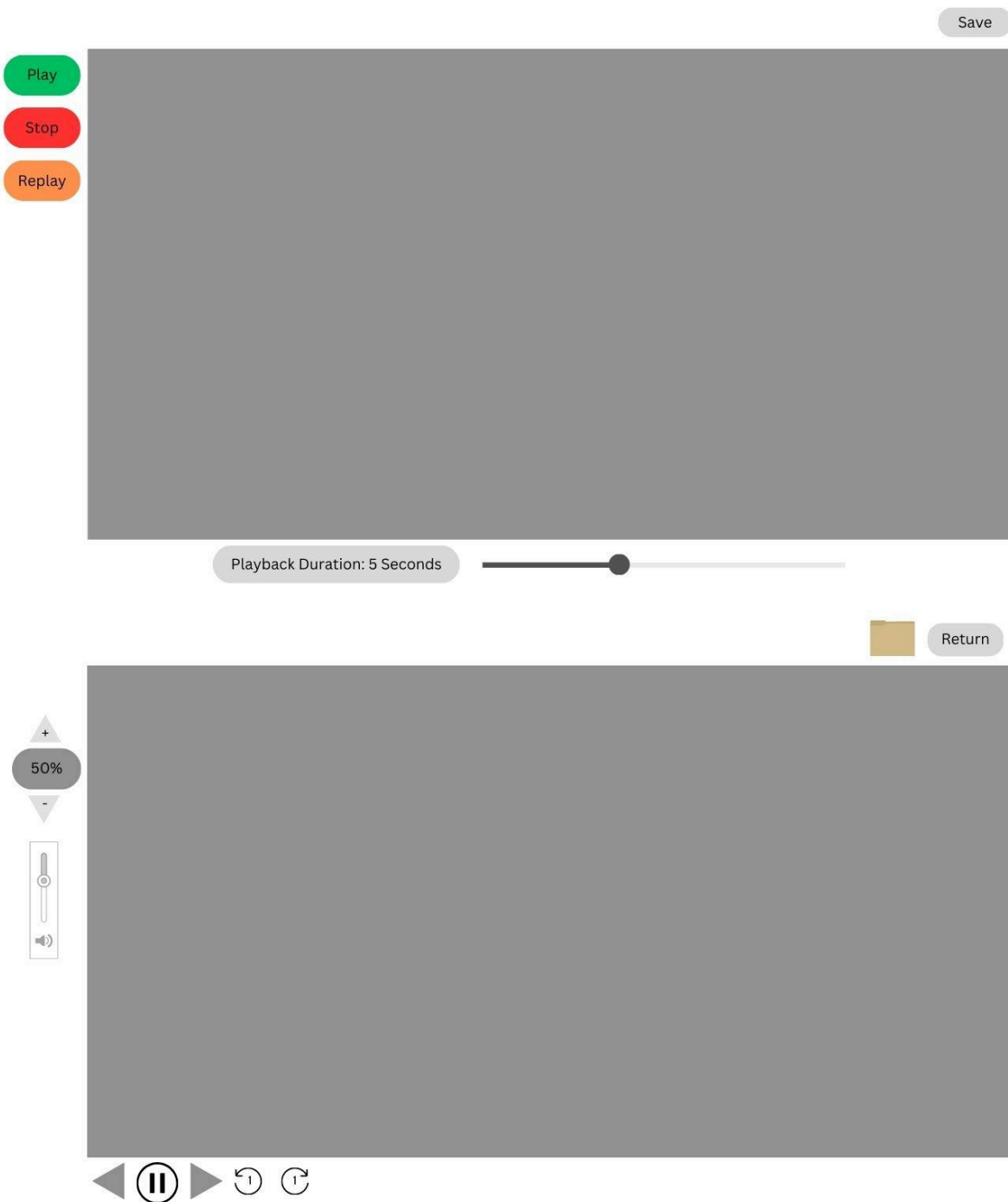


RePoste Fencing Replay System

Table of Contents

1. UI Prototype	2
2. Problem Statement	3
3. Design Explanation	3
4. Implementation Plans	4
5. Potential Improvements	4
6. Client Feedback	5

UI Prototype:



Problem Statement

Our team plans to provide an open-source replay system for the sport of fencing that will run on MacOS and Windows. This system will be capable of recording, replaying, and saving the video for later review. The current recording system used by USA Fencing is closed source, runs on one OS and is expensive for fencing clubs to access. This, in practice, means that the referees and judges are required to use a Mac device that has not been updated past a certain system update, the outdated software has not been updated to support modern OS versions of Mac and limits the possible devices that can be used.

To address this problem the system proposed should be able to run on at least current versions of Windows and Mac. It shall be able to capture video when signaled, replay the footage at selected speeds, move through the replay frame by frame, and capture the video at a minimum 30 frames per second. In the UI Prototype we have included a volume slider to represent the goal of having audio recording as an option for replays.

By developing this open-source tool, we aim to empower fencing organizations and individuals to enhance the quality of refereeing, coaching, and athlete performance.

Design Explanation

The requirements for this iteration are relatively the same as our first concept. Our team has stuck with Python as a means of rapid development and testing. We must focus on implementing a minimal UI to assist with preserving computer resources. A basic video player with options to manually initiate recording, replay, and play/pause features. Any other features, while visible and easily accessible, should not be distracting or get in the way of the video feed.

We chose this design to focus on basic functionality required of the program. Implementing UI elements to start/stop video recording, access replay footage, and saving of replays as mp4. This was completed by using Python and the following libraries:

<https://pypi.org/project/PyQt6/>

<https://pypi.org/project/imageio/>

Prototype Implementation

After the first stage of prototyping we have implemented the functionalities for video recording, replaying, and saving replay footage. This was done by using Python and the select few libraries listed in the previous section. The team had intended to complete more work on the replay functionality, but ran into issues with video playback. These playback issues were resolved by switching from the OpenCV-python library to Imageio. With a solution to that issue we can move forward implementing the replay footage functions. The focus going forward will be polishing the replay function, variable replay speeds, frame by frame analysis, and implementing a variable for replay duration.

Prototyping in Python has helped us more rapidly work towards the base logic of our replay system, as well as help us better understand methods for designing and implementing features if we turn to C/C++ instead.

Implementation Plans

1. Work on polishing the replay functionality.
 - a. Python Library: imageio, PyQt6
2. Implement replay at variable speeds/frame by frame analysis.
 - a. Python Library: imageio, PyQt6
3. Implement variable replay duration.
 - a. Python Library: imageio
4. Add Elements to the UI to change replay speed and video frame.
 - a. Python Library: imageio, PyQt6
5. [Get Feedback from Terry]

Potential Improvements

To enhance the user experience, several areas could be explored:

- *Intuitive Layout:* Prioritize a clear and uncluttered interface to reduce the learning curve for less tech savvy users.
- *Optimize button placement:* Position buttons to minimize confusion and improve usability.

- *Visual Clarity*: Ensure the video remains the primary focus, with secondary controls easily accessible.

Client Feedback

1. Is the UI design an improvement on the previous iteration?
2. What other additions would you like to have on the UI?
3. Is there anything you'd like removed from the UI?
4. Is there anything essential that we might be overlooking given our lack of fencing knowledge?
5. Any suggestions on potential improvements on the UI?