SE 492 - Report 03
Decision Support in Racket Games
Status Report 3

2/13 - 2/27
Group Number: SDMay20-44
Project Title: Decision Support in Racket Games
Client: Simanta Mitra
Faculty Advisor: Simanta Mitra

Team Members

Benjamin Kramer
Brian Guidarini
Katelyn Sinn
John Rachid
Christion Barnes
Aiden McMinimy

Biweekly Summary

This week concluded our first official iteration of the semester. The MVP web API is fully implemented now, as is the MVP for the Python processing component of our application. The MVP for the front-end is not yet finished.

Accomplishments

- Ben
    - Came up with a system to optimize the speed of video processing by using motion detection on the same video frame for player finding and ball finding. The system also showed a simple way to determine more information about the status of the rally. However, after accuracy testing, it was determined the system starts to take major accuracy losses the longer the video goes on. Advantages and disadvantages are being weighed and thought through to find a new way to keep optimization gains but retain accuracy.
    - Did major refactoring, making video analysis far more straightforward and less confusing.
    - Began testing new methods for ball identification to bring accuracy back up. Going forward, the worst case scenario is that the speed gains will have to be dropped, as accuracy is reliably strong using old methods.
- Brian
    - Finished up the MVP web API. This includes endpoints for user creation, video retrieval, video processing, and video data collection.

    - Created job objects that represent a processing job. It includes data such as the processing status, the video that needs to be processed, its directory, the user who created, etc…
    - Created the Submit-Job endpoint which is capable of running jobs in parallel rather than one at a time. This results in a much more efficient use of resources, since about 5 jobs can be processed without any loss in the frame rate (on my system).
    - Extracted most of the code in Main into a new VideoProcessor object so it plays well with multi-threading.
    - The web API is now fully integrated with the front-end. Videos are processed, and json data is stored. With the Get-Splices endpoint, this data can be retrieved for any given processed video.
- John
    - Created hit recognition that is used for creating timeslices
        - Added two helper methods to help reduce the number of false positives
        - Not fully fleshed out might there are some false positives and false negatives
    - Updated main to create lists of volleys that will be sent to the front end
- Aiden
    - Created the header for the React application
    - Created the first iteration of the recommended view
- Katie
    - Created store to house volley info
    - Experimented with ajax calls frameworks, set up axios
    - Experimented with video frameworks, used video.js
    - Connected to backend endpoints, still getting them to work
- CB
    - Created a video uploading progress screen
    - Screen displays percentage completed of a video file being uploaded to our web app.
    - Research different styling techniques to make screen look more professional/modern
- All
    -

Pending Issues

- Optimization is needed for the Python image processing
- Several front-end components from this iteration remain unimplemented

Time

Team Member	Bi-Weekly Hours	Total Hours
Benjamin Kramer	13	41
Katelyn Sinn	11	28

Brian Guidarini	30	56
Christion Barnes	13	30
John Rachid	15	38
Aiden McMinimy	10	26

Upcoming Tasks

- Ben
    - Work with John to finally pull everything that has been learned and tested for video processing so that all the information needed from the processing is collected and focus can be shifted to other areas of the development like the front and back end.
    - Polish ball identification to limit false readings.
    - Start looking into and potentially contribute to the front end.
- Katie
    - Work on video view
    - Get video upload to work
    - Retrieve volley info
- John
    - Work with Ben in order to fully polish hit identification and the python processing
- Brian
    - Jobs will need to provide more detailed status information, such as the spot a job is in the processing queue and estimated completion time for jobs that are currently being processed.
    - An endpoint for health metrics will need to be created. This will provide information such as up-time, the number of running jobs, the number of waiting jobs, and the frame-rate of jobs.
    - Jobs need to be stored in secondary storage in order to preserve statuses after the service either fails or restarts. This will make it so we can restart interrupted jobs so they get a chance to run. This will also make it so users can check the status of jobs that were run before the service restarted.
- Aiden
    - Spend some time improving the components I failed to properly implement
    - Help Katie with implementing the React App
- CB
    - Begin developing other pages that the users will see and interact with on the web app
- All
    -

Advisor Meeting Summary

Our advisor was out of town for our meeting this iteration, so we will meet this week.