Memo – Functional Test Report

**To:** Professor Pisano, Professor Kia

**From:** The Karty Boyz

**Team:** 15

**Date:** 4/25/14

**Subject:** Functional Testing Report

# Introduction

Our functional test aimed to show the cohesive functioning of our project overall, testing that the each piece individually functioned as we intended, as well as testing that the pieces interact correctly. Specifically this means testing the process of data collection, processing, and analysis shown in the workflow of recording a video, uploading it to the website, and querying the resulting data. To do this we must also demonstrate that the each component: the website, the processing suite, and the querying system are working as designed.

# Equipment and Setup

The equipment used in this test is fully described in the test plan. The primary local equipment used for this test was a Nintendo 64 with Mario Kart 64 and a USB composite video capture card with a custom microphone module, and a computer with a web browser. In addition, several virtual machines running on Amazon Web Services (AWS) were used for hosting the database, website, and video processing software. In addition to this equipment, we also had a pre-recorded video file that was used for processing the video.

# Measurements Taken and Evaluation

As the test spanned the system, several measurements were taken at each stage of the test. Boolean results for if the transitions between components functioned correctly were also treated as measurements.

## Video Recording and Audio Capture

To verify that we are able to record ambient speech audio independent of the game audio and test that the microphone module works, a sample video was taken with audience volunteers.

The audio track of the video was analyzed for the following criteria:

* One of the audio channels has the sounds from the game
* The other audio channel has audio from the environment
* The audio from the environment is free of noise
* The environment audio can be understood if played from a computer speaker

During our testing we verified that a laptop connected to the capture card could record both of these audio channels correctly and we verified that all of the above criteria were met by examining and listening to each channel of the audio independently. This confirms that our microphone module is working correctly.

## Website

In order to assert that a user is able to effectively use the system, a test of major website functionality was demonstrated.

For the website to be considered usable these conditions must be met:

* The user can submit a video file for processing
* The user can download processed race videos for viewing
* The user can submit custom queries
* The user can view the result of their queries

Our test determined that our website meets these requirements. We successfully uploaded a video, downloaded and watched a video, and ran sample queries. We also demonstrated the degree to which the website exceeded these criteria by showing that uploading a video produced a progress bar, and that videos could be streamed from inside the browser.

## Video Processing

Because extracting event information is such a critical part of this project, rigorous testing of the video-processing suite was required.

To be accepted, this component must:

* Accurately detect beginning and endings of races in a user’s video
* Accurately detect events in each detected race

The first criteria was shown to work last semester during our initial testing, and only briefly considered during this test. The second criteria was evaluated through a unit testing suite that verified that each event detector in the suite correctly detects events on frames of video depicting the event, and correctly fails to detect events on frames not depicting those events. This testing suite contained 81 tests, of which 81 of them passed meaning this component is considered functional.

## Audio Processing

Our clients were very interested in being able to use speech recognition to detect utterances of the phrase “Tag It” in the ambient audio. Unfortunately, due to last minute complications with the used speech recognition techniques only the feasibility of a new approach could be demonstrated.

To determine this approach’s feasibility a test was conducted recording a sample “Tag It”, and evaluating if the system’s interpretation of the audio was in a list of phrases to be accepted as “Tag It”.

During the test, we demonstrated that the phrase “Tag It” was recognized by the alternative system as “Target” which matched one of the approved interpretations. This means that the alternative speech recognizer is considered feasible for the final product, though some of our prior results were mixed. We are cautiously optimistic about this result.

## Query Language

The data generated by processing audio and videos is not particularly useful on its own. One has to formulate questions that they would like to attempt to answer using the data. To this end a terse way of expressing questions is required.

To be useful the query language needs to:

* Be more terse than SQL
* Be transformed in to efficient SQL
* Produce results for a set of sample queries
* The sample query results must have the correct number and types of columns
* The results of these queries must match the results of a hand-written SQL version

During the test, 5 sample queries were demonstrated. For each query, it was shown that the output columns in the result set matched one of the outputs specified in the query. It was also verified that the SQL generated by the query language was equivalent to what one familiar with the schema would write by hand. As a result, all of the above criteria were met and the query language is considered passing.

## Integration

With each of these components working individually we wanted to make sure that each component interacted successfully with the rest of the system.

For the system to be integrated it must do the following:

* Create a record in the database for a new session when a video is uploaded
* Notify the video processing system that there is a new session to be split
* Create a record in the database for a new race when a session is split
* Notify the video processing system that there is a new race to be processed
* Create a record in the database for an event when one is detected in a race
* Store all uploaded videos and split videos persistently

During the test we showed the full workflow. We uploaded a session video, and verified that it was recorded in the database, saved persistently, and that the video-processing suite immediately began to process this video. When that finished verified that it saved the detected races to the database, and persistently stored the split videos. We then verified that the video-processing suite began processing the race that had just been split. When this processing was complete, we verified that there were event objects in the database. After verifying all of this we were able to conclude that the integration of the above components was a success.

# Conclusion

As a result of the above evaluations we are able to conclude that we will be able to deliver a functional completed product.

While certainly not polished enough to be considered a finished project, this test indicates that we have by and large succeeded at our goals. Our current concern is now integrating the Google Speech API with our current audio processing code, though this is underway. The rest of our work is now better integrating our codebase with AWS, and trying to improve the user experience of our query language.

Overall we are optimistic about the outcome of this project, and believe that this test proves we have built a functional product.