Senior Design

ENG EC 463

Boston University Electrical & Computer Engineering

To: Professor Pisano, Professor Kia
From: Michael Abed, Jonathan Bell, Johan Mickos, Josh Navon, Mark O'Brien
Team: #15 The Karty Boyz
Date: 11/12/13
Subject: First Deliverable Testing Plan

1.0 Overview

1.1

The first testing deliverable aims to demonstrate the integration of video processing, database, and web server. Specifically, we will exhibit the video-splitting module and its interaction with the database. Correct execution of the test will result in one Mario Kart 64 session video being split up into separate videos for each race it contains. These videos will be placed in a specified directory where the web server can process requests regarding individual races. The full test will consist of a user uploading a pre-made test video, the processing software extracting individual races, and the storage API generating video files for each extracted race.

1.2

In essence, this deliverable will show all high-level components of the project working together. It addresses the challenge of integration, a common roadblock in large projects. Successful execution of this deliverable ensures that further testing deliverables will be easier to meet, as all necessary components are already integrated.

2.0 Criteria/Process

2.1

First off, the web server is run, allowing the user to navigate to the upload page. Upon uploading the session video, the first phase of video processing will begin. Phase 0 detects starts and finishes of individual races, packing these data into a JSON structure and sends it to the database in the format:

```
{
    “start_time” : <race start time>,
    “duration” : <race duration>
}
```

The database will call `ffmpeg` to split the original video, specifying the source video, start time, and duration. The resulting race video is then made available to the core API. Data will continue to be sent from the video processing in phase 0 until the entire source video has been analyzed for races.

4.0 Equipment and Setup

4.1

The following setup will be used:

- race_snippet.mov — Test session video containing multiple Mario Kart 64 races
- Linux server — Used to run database, video processing, user interface, and video storage.
  - Database server (from n64_storage_flask repository)
  - Video processing (from n64_img_processing repository)
  - Client interface/control (from n64_core_api repository)
  - Video server (web server at time of testing)
- Client web browser — Used to upload the test session video to the cloud

5.0 Measurements

5.1

After processing has been completed, we can query the database for specific data regarding the uploaded session and the races it contains. The database will store information about the race start time and duration, as well as a URL where a video file for the race can be downloaded from. This video file will contain only a single Mario Kart 64 race extracted from the original game session.

5.2

In order to verify measurements, we will compare the generated race videos to the test session video. By comparing the race sequence number to the race position in the video, we will be able to verify correct execution of the test. Similarly, we will compare the data contained in the database to timestamps in the test session video to verify accuracy. Finally, we will cross-reference the number of races created in the database with the known number of races in the test session video.

6.0 Conclusions

6.1

** *Note: This section is a summary of Criteria/Process and Measurements* **

After the test has completed, we will have the following data available to us:

- Database tables and schema
  - Sessions

```
{
      “date”: <upload date>
      “video_url”: <location of session video>
      “video_split”: <True/False>
}
```

  - Races

```
{
      “session”: <parent session>
      “start_time”: <start time in parent video>
      “duration”: <length of race video>
      “video_url”: <location of race video>
      “race_number”: <race # in parent video>
}
```

- Individual race videos formed after splitting
- Console outputs for database

In order to conclude the test as successful, we will compare these data to the expected outcomes. The database will contain an entry for the newly uploaded session, as well as entries for each individual race contained within this session. The number of new entries in the Races table will be compared to the actual number of races in the session video. The files created during the splitting process will be inspected to verify that they do in fact represent individual races. The console output of the database can also be analyzed to verify that the correct number of POST and GET requests were generated.