Boston University
Electrical & Computer Engineering
EC463 Capstone Senior Design Project

Problem Definition and Requirements Review

N64 Analytics Engine

Submitted to

Dan Ryan
Bytelight
183 State St.
(617) 828-6218
dpryan87@gmail.com

By

Team 15
Karty Boyz

Team Members

Michael Abed mgabed@bu.edu
Jonathan Bell jbell@bu.edu
Johan Mickos jmickos@bu.edu
Joshua Navon navonj@bu.edu
Mark O'Brien mark0b@bu.edu

Submitted: 11 October 2013

Customer Sign-Off _____

1.1.1.1 N64 Analytics Engine

1.1.1.2 Table of Contents

2 Project Summary

This project aims to create a system for Mario Kart 64 analytics. The project includes processing recorded Mario Kart 64 videos, collecting statistics about events that happen during gameplay, and presenting these statistics to the user in an intuitive, useful, and fun format. This will include allowing the user to view race videos, screenshots, and ask questions about race events. In addition, the user will be able to provide a recorded commentary of their gameplay to be processed for automated tagging of user-specified times and automatically detected exciting race sequences.

3 Need for this Project

Mario Kart 64, like several other retro games, has stood the test of time. Mario Kart 64 has been a favorite of many since it was released. Therein lies a problem, though. The life of the modern Mario Kart 64 gamer has turned into a dismal, meaningless existence. Never lacking worthy opponents, skilled players are left helpless, unable to stroke their egos as they fruitlessly insist upon their dominance.

Few have tackled the issue of gathering data and statistics from actual Nintendo 64 (N64) game-play, presenting us with an exciting opportunity. The Karty Boyz will make it possible to gather race and session statistics for Mario Kart 64, enabling players to back up their smack-talk with cold hard stats. Furthermore, this project aims to answer some of the mysteries of Mario Kart, such as, how random the item generation sequence is, to what extent does place affect the item received, what is the probability of saving oneself from a banana spin-out, etc. Such information will prove useful to enthusiasts who can use it to improve their gameplay. With the ability to also tag events with speech, exciting moments will never be forgotten, providing an even more immersive experience. The finished product will give the devoted retro-gamer a clearer understanding of the game, a medium to better formulate strategies, a fun and interactive post-Mario-Kart experience, and a way to prove lofty claims of superiority.

4 Objective and Deliverables

4.1 Objective

The objective of this project is to provide an enhanced Mario Kart 64 experience. This enhancement comes in the form of the ability to retroactively inspect and analyze races for interesting statistics and exciting moments. To do this, we aim to collect Mario Kart 64 race footage, detect events in the footage, and provide an interface for asking questions about those events. In addition, because it is often desirable to record when exciting moments happened, we aim to allow a user to tag these moments vocally through either manually with some key-phrase or automatically by analyzing background volume levels.

In addition to the portions of the project that the user will interact with, we have several technical objectives. We aim to leverage state of the art computing paradigms and tools to create the best product possible. To accommodate this, the computation for this project will be done on the Massachusetts Open Cloud (MOC), which will also host video files and the database. Algorithms for processing the videos will be designed using OpenCV.

4.2 Deliverables

- A set of algorithms for extracting events from Mario Kart 64 race videos
- A persistent database of events gleaned from processing videos
- An interface for uploading videos to be processed
- An interface for querying events for specific statistics
- An algorithm for detecting specific phrases and elevated volumes
- A microphone module for recording race commentary and audio events

5 Visualization

5.1 User Facing Components

The following figure provides a high-level overview of what will be visible to and end-user. The primary components that they will interact with will be the video uploader and the database query interface.

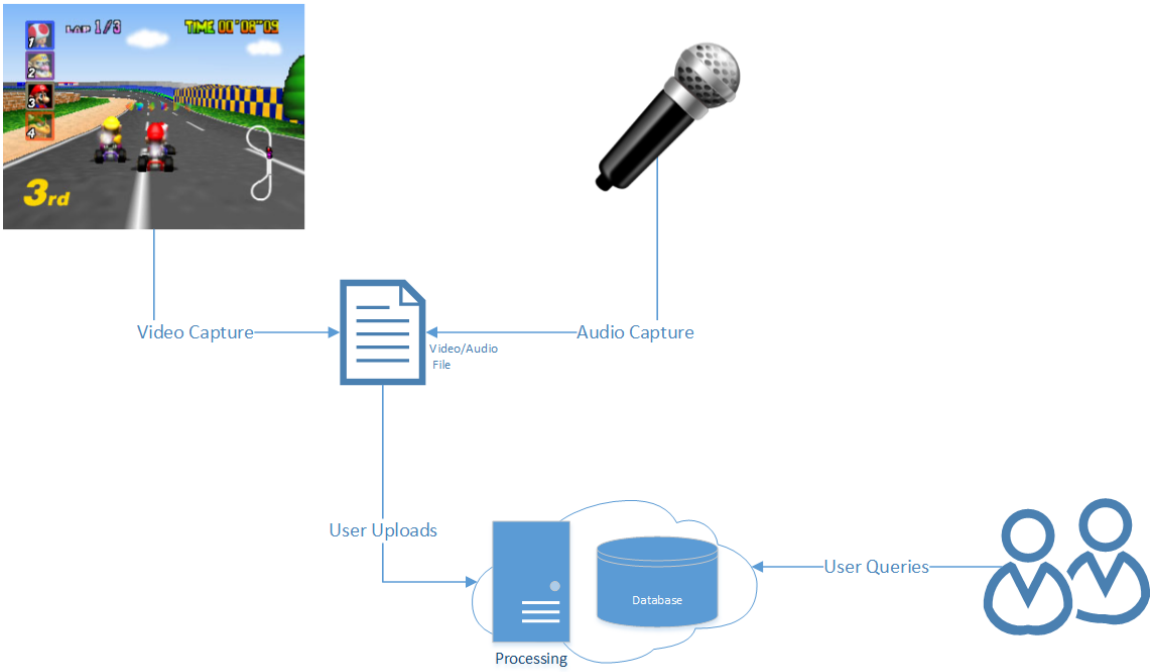

Figure 5.1: *Video and audio are captured to a standard video file format. This file is uploaded to a server for processing, which breaks the video into its component races, and stores the events in a database. Users are then able to query the database for events and videos.*

5.2 Cloud Processing Components

While the user will not see very far into the cloud processing that happens, they will still be exposed to its workings to some degree. The following figure shows the processing, as a user of the application should view it.

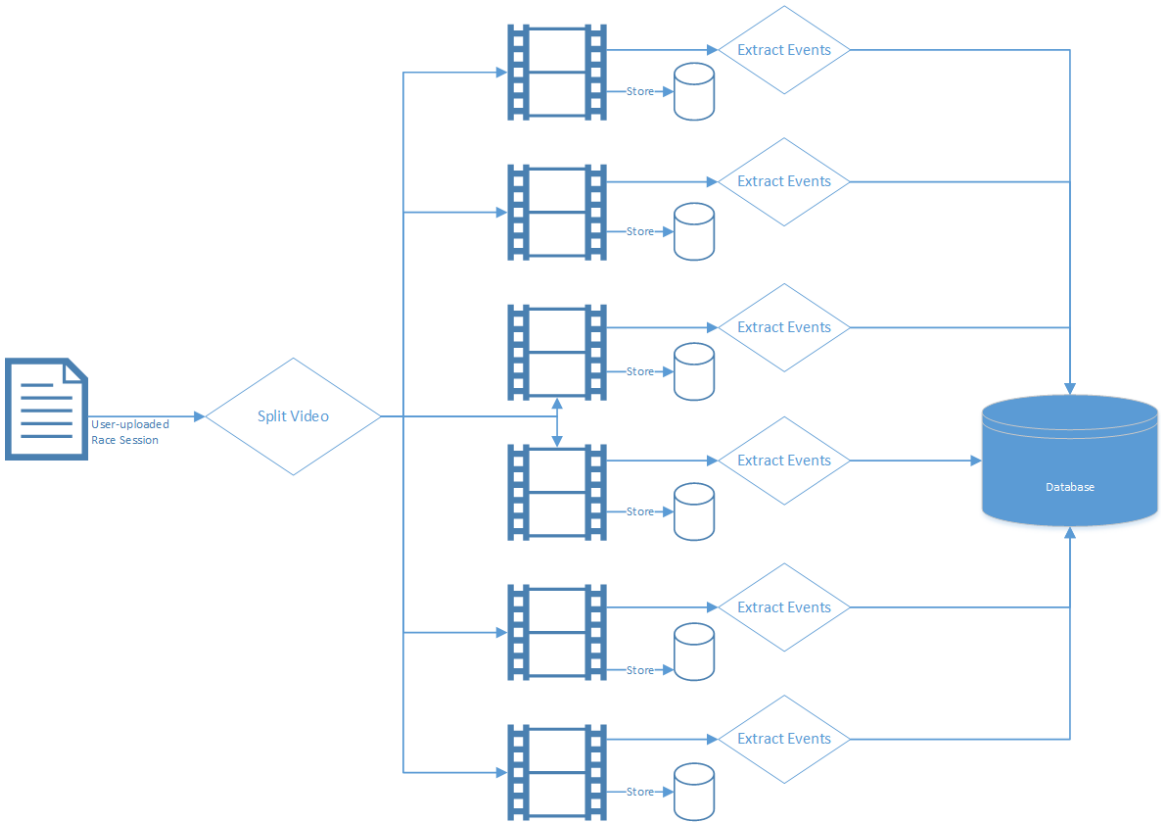

Figure 6.2: *Once the user uploads their video, it is split into its constituent races, which are each stored and processed for events. These events are recorded in a database.*

6 Competing Technologies

In our review, Joyent's Kartlytics was discovered. Kartlytics is a website offering Mario Kart 64 analytics as a demo for a database product that Joyent sells. It takes the form of a website that users can visit and view pre-computed information about past races (played by Joyent employees). For each of these races, they track several variables including position over time, lap times, number of players, and item distributions.

Figure 6.1: *Screenshot of Kartlytics homepage showing which tracks get played often, 4 classes of race they've calculated, and item distributions.*

All of the features included in Kartlytics will be encompassed in our product in some form. Since our product aims to allow for general analytics queries, Kartlytics can be viewed as a static view of our project's output.

7 Engineering Requirements

7.1 Event Detection

The most crucial requirement for this project is event detection. In Mario Kart 64, there are hundreds of events occurring every minute that define the current state of the race.

7.1.1 Types of Events to Be Detected

1. Changing places of characters
2. Beginning and end of a race
3. Pausing of a game
4. Lap changes
5. Collisions
6. Acquiring items
7. Shortcut detection (specific to Koopa Troopa Beach map)
8. Boost detection (starting boost, drifting boost, and turbo boost)

7.1.2 Detection Accuracy

1. When an event of the above types occurs, it must be detected with at least 90% accuracy.
2. No more than 5% of the detected events may be erroneous.
3. Race separation must be performed with 100% accuracy.

7.2 Audio Event Tagging

As part of processing the detected events mentioned above, the engine must also tag these events according to their type. This information will be stored in the database for later retrieval. Additionally, there will be an event type "exciting moment" for capturing events not explicitly specified by the game. We will capture these exciting events in two ways.

7.2.1 Key Phrase Recognition

1. The phrase "Tag It" must be detected.
2. False positives for detection may be no more than 15% of the time.
3. False negatives for detection must be no more than 3% of the time.

Because it is much more important to not miss user-specified tags, a relatively high false-positive rate is acceptable for this requirement.

7.2.2 Volume Level Detection

Because elevated ambient volume is often a sign of excitement occurring in the game, it will be used as another indicator for moments that should be tagged.

1. Background volume surpassing some threshold (to be determined) must be detected.

2. Detection must be relative to ambient volume, not an absolute threshold.

7.3 Database

A persistent database is required to store events. These events must be accessible to general user queries.

7.3.1 General Requirements

1. Must allow for storage of all types of events described above.
2. Must allow for users to query the events without using SQL.

7.3.2 Specific Questions

The following questions must be answerable using the query system.

1. How many times did Toad get a red shell during today's races?
2. What is my best, worst, and average lap time on Koopa Troopa Beach?
3. How many times did Yoshi get 2 lightning bolts during a race and come in first?
4. How many times did I get passed after getting hit by a red shell?

7.4 Cloud Services

In addition to the database, a number of general computing tasks are required. These include video streaming, processing, and storage.

7.4.1 Video Services

1. User must be able to upload race videos.
2. User must be able to download and view race videos after uploading.

7.4.2 Video Processing

1. Uploaded videos must be split into individual races and stored persistently.
2. Races must be processed for all events described above.
3. Processing of a race must take no longer than twice the race duration.