High Level Design

Group 8: Sudden Death

Group members: Yomara Acevedo, Annie Conover, Alex Jones

Table of Contents

1 Introduction **2**

2 Problem Statement and Proposed Solution **3**

2.1 Problem Statement 3

2.2 Proposed Solution 4

3 System Requirements **4**

4 System Block Diagram **5**

We are including the entire flow of information in our block diagram for clarity, rather than only including the devices we will build. 5

4.1 Overall System 5

4.2 Subsystem and Interface Requirements 7

4.2.1 Gamecube internal memory 7

4.2.2 Gamecube EXI bus 7

4.2.3 Memory card breakout fixture 7

4.2.4 Microcontroller 7

4.2.5 Wi-Fi Transmitter 7

4.2.6 Central Hub 8

4.2.7 GUI 8

4.3 Future Enhancement Requirements 8

5 High Level Design Decisions **8**

5.1 Memory Card Breakout Fixture 8

5.2 Microcontroller 9

5.3 Wi-Fi Transmitter 10

5.4 Central Hub 10

5.5 GUI 10

6 Open Questions **10**

7 Major Component Costs **11**

8 Conclusions **11**

Bibliography **12**

1 Introduction

We currently utilize advanced sports metrics because they provide in-game productivity and measurements. They allow us to predict player performances, highlight strengths and weaknesses, and in general provide objective ways of consuming sports media coverage.

There is an increasing interest in eSports (that is, electronic sports). League of Legends and DotA 2 are two massive online multiplayer video games with millions of viewers and participants around the world.[1] Both of these games have been updated to include an in-client way to watch competitive matches that shows live statistics in an infographic in order to enhance the viewing experience.[2] Additionally, there is an even more in-depth version of this available to commentators so they can provide the best analysis possible.

Super Smash Bros Melee currently has no in-client method for tracking statistics. Since there is no means to see statistics as the match happens, viewers do not have a good method for digesting the match in front of them through relevant statistics, and commentators cannot offer the highest quality of analysis possible. Despite this need, Nintendo has no desire to patch a game that is operated on a console that is 15 years old--almost three generations old.[3]

In fact, Melee currently has a thriving and ever-growing competitive following even with Nintendo's disinterest in its own game.[4] The Evolution Championship Series (EVO) 2016 SSBM tournament had 2,350 entrants and a peak of 232,900 viewers on the video game broadcasting website Twitch.[5] EVO has been televised on ESPN2, with roughly 2 million unique viewers and 201,000 peak concurrent viewership for EVO 2016 finals (not including the Twitch numbers).[6] These numbers for each platform are comparable to a regular season baseball game broadcast on the MLB network, which averages 260,000 viewers for regular season games.

Although Major League Baseball games broadcast on other networks typically garner more viewers, the comparison between two dedicated channels like Twitch and the MLB network demonstrates incontestable interest in a specific sport, or in this case, eSport. While Melee is an order of magnitude below major sports in viewership, it is clear that the community is not to be ignored.

---

[1] *theScore esports*. LoL News. https://www.thescoreesports.com/lol/news. Accessed 11 Dec. 2016.
[2] *LoL eSports.* VODS. http://www.lolesports.com/en_US/vods/all-star/all_star_2016. Accessed 11 Dec. 2016
[3] Gordon, Justin "Adaptive Trigger." "Why Super Smash Bros. Melee HD (with Online) Would Be A Huge Success for Nintendo" *EventHubs*. 23 Oct. 2016. https://www.eventhubs.com/news/2016/oct/23/why-super-smash-bros-melee-hd-online-would-be-huge-success-nintendo/. Accessed 11 Dec. 2016.
[4] Groot, Justin. "EVO 2016 Might Be the Most Competitive Smash Tournament Ever" *Kill Screen*. 15 Jul. 2016. https://killscreen.com/articles/evo-2016-most-competitive-smash-tournament-ever/. Accessed 11 Dec. 2016.
[5] Cuellar, Joey (MrWiz) "Evo 2016 Numbers - SFV 5065, Smash 4 - 2637, Melee - 2350, Pokken - 1165, GG:XrdR - 903, UMvC3 - 770, MKX - 707, T7 - 543, KI - 540 #Evo2016". 1 Jul 2016, 12:08 AM. Tweet.
[6] "Evolution Championship Series News and Updates." *ESPN*. 21 July 2016. http://www.espn.com/esports/story/_/id/17057785/catch-evolution-championship-series. 11 Dec. 2016.

2 Problem Statement and Proposed Solution

2.1 Problem Statement

Since Melee has no built-in way to see match statistics as a competitive match happens, viewers and commentators alike are hindered. This is the case because the statistics recorded by the game engine are mostly trivial--who killed who, how much damage a character dealt total--and are only presented at the end of a match, as can be seen in Figure 1 below.

There is also a list of small “awards” handed out to each player. These clearly involve a game functionality of keeping track of statistics such as whether a player used a certain move repetitively or survived a long time when close to death, but there is no way in the game to see the detailed statistics behind these awards. If this information is being tracked, there must be a way to extract and interpret it.

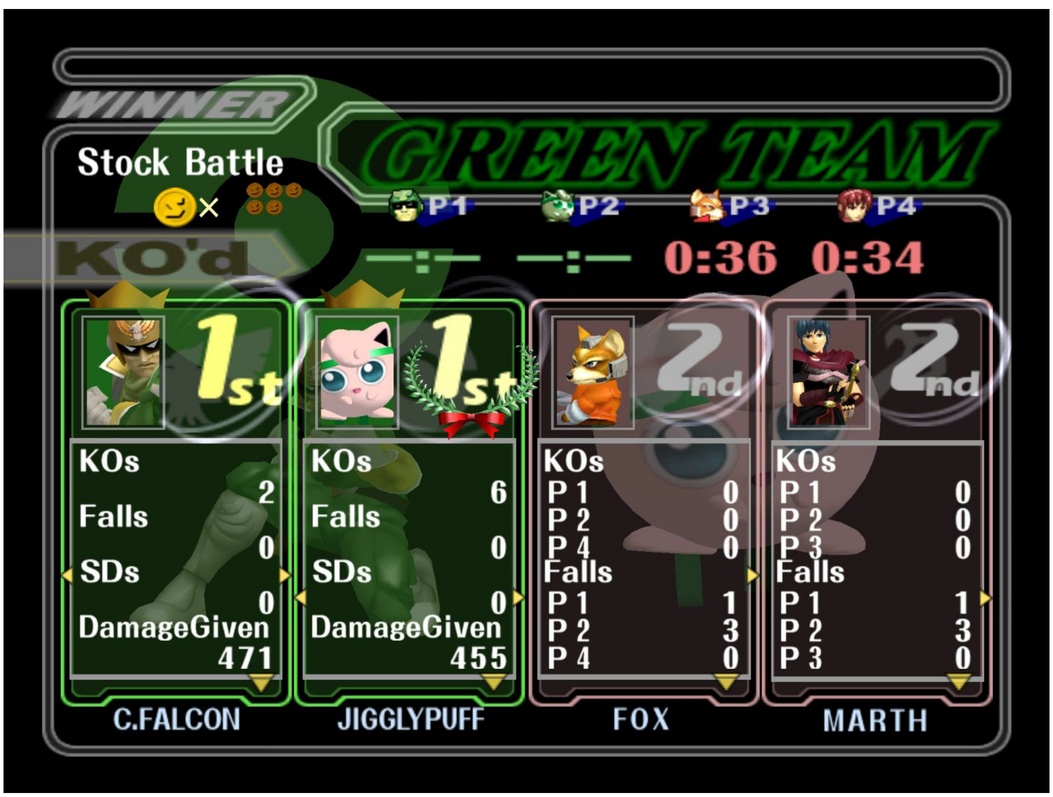


Figure 1. Results of one match in SSBM.

2.2 Proposed Solution

Our understanding of how a GameCube games works is the following: There are two inputs in the form of the game disc and the user controller. The GameCube microprocessor interprets these inputs and provides three outputs: a visual form, audio, and memory. We want to grab information that is being temporarily sent to memory during the match and interpret it.

Jas Laferriere, a user on SmashBoards.com better known by his tag name “Fizzi,” posted about a year ago that he utilized a breakout board that can be inserted into the the memory card slot to interact with an SPI-like device in a Wii in order to read the information that the game is transmitting in real time.[7] It measures the electrical signals that the game is transmitting through the microprocessor of the Wii.

For our project, we want to create a similar device that extracts information in a similar way to “Fizzi”, and then outputs the information on a Wi-Fi chip so that it can be viewed live on a website. We will have to then interpret the data and display it in a user-friendly format. We will be using the GameCube instead of the Wii since it will be more relevant to competitive Melee tournaments because most tournament matches are played on GameCubes.

3 System Requirements

The most basic requirement for our project is the shape and size necessary for it to be used. The final board we print must be the correct shape to plug into a Gamecube memory card slot, while also being lightweight and small enough that it will not sag while hanging out of the slot (realistically probably a maximum of a few ounces). Ideally, it would be small enough to fit into a gutted memory card casing, but that is unrealistic. Instead, we aim to print a board as small as possible and then 3D print a plastic casing that mounts it properly into the memory card slot.

Ease of use is a high priority for the design of this project, so the goal is for the chip to be “plug and play”. This means that to be used, our chip only need be plugged into the memory card slot, and it would do the rest of the work from there. Additionally, for the “internet of things” scaling of this project where many chips plugged into Gamecubes all put data in one database, we will likely need a central hub, which will require setup and monitoring by an overseer. In this case, the ideal scenario is that our hub system could support somewhere upwards to fifty devices, which is about the maximum number of Gamecubes used at once in the early stages of the largest tournaments. That is a bit more of a stretch goal, and even the ability to support roughly five systems at once would be sufficient to track the matches of all MIOM top 100 ranked players (the Melee equivalent of the AP poll) at a given tournament, which is where essentially all the demand for statistics lies. Most tournaments take place in a ballroom-type space, so to support a tournament-wide network of devices, the supported range would need to

---

[7] Laferriere, Jas. "Statistics In Melee - The Tech (Part 1)." *Melee It On Me*, http://www.meleeitonme.com/statistics-in-melee-p1/, 29 Sept. 2015. Accessed 11 Dec. 2016.

be 100-200 feet for the largest tournaments. Once again if we scale down to only track top level players, the range would only need to be 50 feet or even smaller if the hub is well-positioned.

The ultimate user interface we have in mind right now is a MATLAB GUI that would interpret database information and display it for a user. The database information will come from a Hadoop system on a Raspberry Pi that will act as the hub for all the Wi-Fi communication.

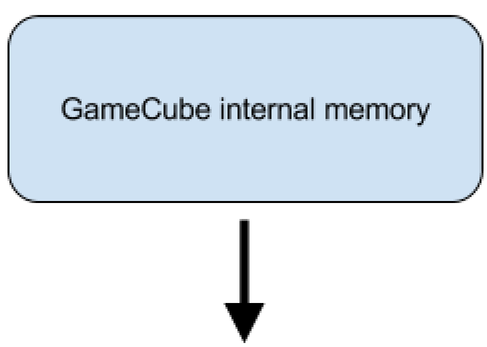
The embedded intelligence will need to interpret the data on the EXI interface. It will have to accomplish this without modifying the game in any way, as modifications of the game are not accepted in the competitive community, so the more invasive the project is, the less likely it will actually be used. This data will be output from Gamecube memory in the form of assembly code, so our program will need to be able to interpret this assembly code. It will need to read all of the assembly code, looking for triggers related to the relevant data we desire, and format text string output that can be passed over Wi-Fi. It is important that all irrelevant game data is weeded out on the chip that plugs into the memory card slot before being transmitted over Wi-Fi, so that we minimize the amount of data that needs to be sent. The embedded intelligence on the hub chip will need to be able to read data from the Wi-Fi receiver and output it through USB to a text file.

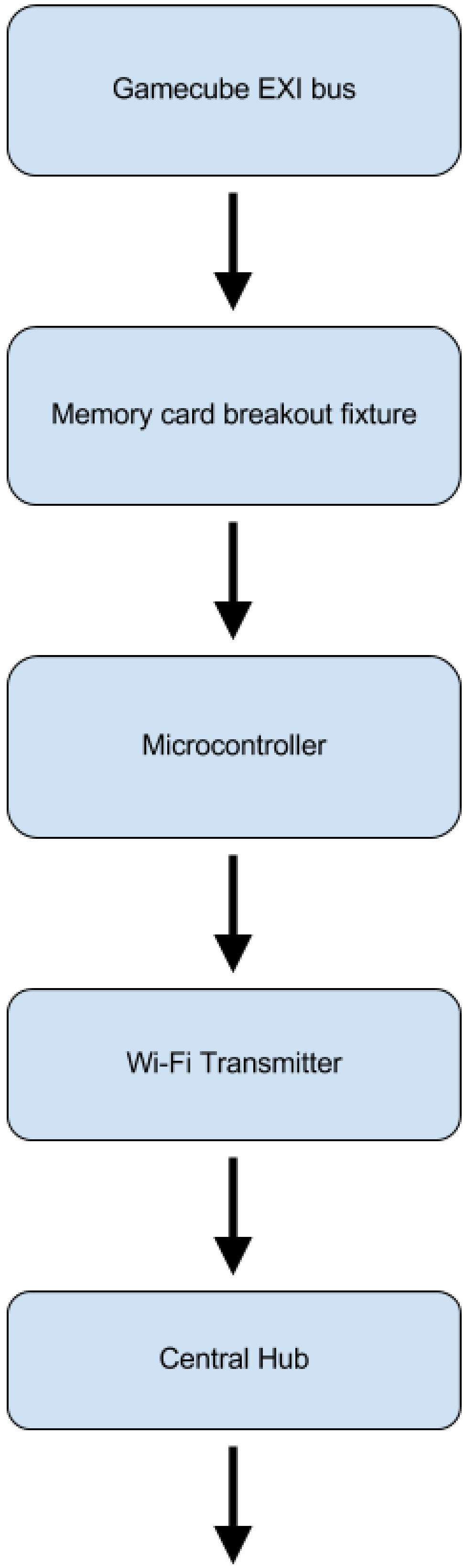
Since the board would be operated in conjunction with a GameCube, and does not need to be portable or mobile, it can be plugged into the wall with a DC power jack. The memory card pin operates at a 3.3 V range, and the DC power jack can certainly be regulated to this.

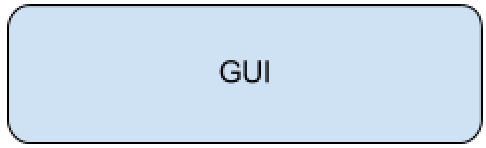
4 System Block Diagram

We are including the entire flow of information in our block diagram for clarity, rather than only including the devices we will build.

4.1 Overall System




Gamecube EXI bus
Memory card breakout fixture
Microcontroller
Wi-Fi Transmitter
Central Hub



4.2 Subsystem and Interface Requirements

The requirements of each subsystem or major interface are described here by subsystem. These lower level requirements support the overall system requirements. Note that major interfaces (such as a wireless interface) should be described like any other subsystem. Don't forget that there will be software as well has hardware in many of the subsystems, and that software will have requirements.

4.2.1 Gamecube internal memory

The Gamecube internal memory is constantly updated with large amounts of information relating to the state of the game. This is where all the relevant data relating to the statistics originates. We simply need this to do its job recording and sending the information

4.2.2 Gamecube EXI bus

A large portion of game data is passed from the motherboard to the memory card slots. This data is what our device will be tapping into. This EXI bus is very similar to an SPI, so will take a slight bit of tweaking from our previous work with SPI to communicate with it.

4.2.3 Memory card breakout fixture

This simply needs to fit properly into the memory card slot in order to tap the pins.

4.2.4 Microcontroller

This microcontroller will allow us to pull the statistics we want from the EXI bus. It will get input data from the memory card pins and process that assembly code to find meaningful statistics. It will need to be connected between the memory card slot and the Wi-Fi transmitter, and will need software on board to process the information. It needs to interact with the 3.3 V pins of the memory card.

4.2.5 Wi-Fi Transmitter

The transmitter will need to take input information from the microcontroller which will be the statistics information picked out by the microcontroller. It will then need to transmit this information to the central hub that puts together the database.

4.2.6 Central Hub

The central hub will need to receive a large data set from multiple Wi-Fi transmitters that can be output to an external device, such as a computer. With that, the data must be collected in formatted files that statistical analysis can be done on. The latency of the system should be minimized in order to keep delay to a minimum.

4.2.7 GUI

The GUI will need to display the information in a readable as well as visual format. In order to be user friendly, the user should be able to select what information will be displayed, filtering by things such as time frame, player, character, type of statistic, etc. Delay should be minimized in order to be used by analysts and commentators in tournament settings and as close to real time applications as possible.

4.3 Future Enhancement Requirements

One enhancement could be availability of the data to all spectators immediately. With our current design, the data will only be updated live on the computer hosting the central hub. It would massively enhance viewer experience to be able to play with these statistics as they watch matches. This would require much more database and computer science skill, and may require different hardware at the hub end than the Raspberry Pi we currently plan to use.

Another possible enhancement is to branch this device's capabilities out to other games. This would involve learning the assembly code of these respective games and including the interpretation of their code in the microcontroller software. This may require more space on the microcontroller, or it may require releasing a different edition of the device for each game.

5 High Level Design Decisions

5.1 Memory Card Breakout Fixture

We know the pins of the memory card, shown in the figure below:

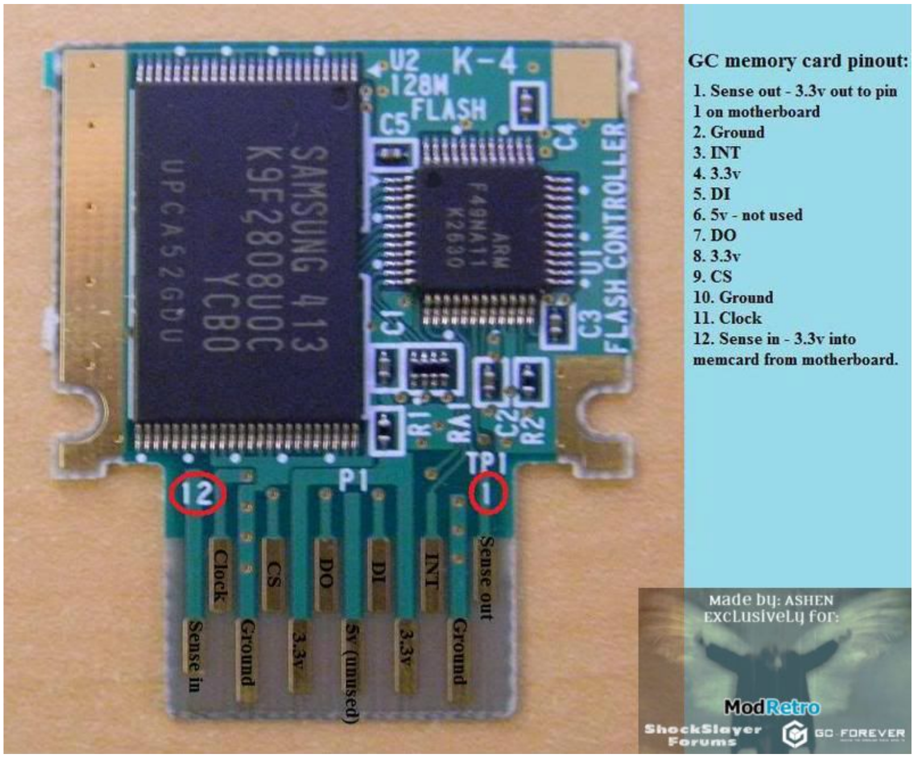


Figure 2. GameCube memory card pinout

We believe the pin associated with reading the EXI bus is 12, Sense In, and writing to the EXI bus is 1, Sense Out.

For testing and development, we will be using a breakout board to interact with these pins, but for the final product the board will plug directly into the slot to interact with these pins. There is a simple size and shape design requirement here that our final chip must fit in the memory card slot. Since it is unrealistic that our board will be the exact same size as a standard memory card, we will custom 3D print a case that can be inserted into the memory card port and hold our board. This can easily be done using SketchUp.

5.2 Microcontroller

This microcontroller will allow us to pull the statistics we want from the EXI bus. It will get input data from the memory card pins and process that assembly code to find meaningful statistics. It needs to operate around the 3.3 V range and needs to be able to perform the logic necessary to process assembly code and output it to a Wi-Fi transmitter. We think we can use the PIC32 for this, but will consider other microcontroller options when the decision must be made.

5.3 Wi-Fi Transmitter

The Wi-Fi transmitter will need to take input signals from the microcontroller and transmit them to be received by the central hub. We imagine we will be able to use the ESP8266 to accomplish this.

5.4 Central Hub

A Raspberry Pi will host a Hadoop platform to collect our data. We will then take our data storage (HDFS) and data processing files (MapReduce) and export. It will be determined based off the size of the data set whether a Hadoop cluster will have to be created. This will be determined based on creating a system to process with the minimum latency possible. Ideally, there will be a small refresh time so that the package code can be exported and refreshed in the GUI in a reasonable time frame.

5.5 GUI

MATLAB MapReduce based algorithms and a custom script will be run within Hadoop MapReduce. After sorting and managing the dataset, it will be inputted into a GUI that would display it with visual and numerical options. The GUI will be designed so that a user may select what information and criteria the data is displayed by. For example, it should be able to be sorted based off of statistic or player.

6 Open Questions

- Complexity of Hadoop platform; Raspberry Pi vs. Raspberry Pi Cluster
  - Amount of memory
  - Connection to Wi-Fi chip
- Time delay of GUI based off of latency of system
- Statistics available for collection-- based off of analysis of assembly code
  - Scope of data visualization
- Multiple questions about Wi-Fi integration
  - How do we make the Wi-Fi chip and microcontroller communicate?
  - How do we set the Wi-Fi chips to only transmit to the hub, or for that matter how do we even make them send information at all?
  - How do we make sure the hub does not receive Wi-Fi information from other sources?

7 Major Component Costs

Component	**Estimated Cost**
Memory card breakout boards for testing	$40.00
3D Printed faux memory card casing	$5.00 max ($0.25 per gram)
Final circuit board - ● Microcontroller ● Wi-Fi chip ● DC power jack ● Passive components	$50.00
GameCube	$45.00
Raspberry Pi	$40.00
Our souls	Priceless
Total:	**$180.00**

8 Conclusions

We consider our project to be feasible. There are a lot of resources available that can get us started, including an in depth proof of concept, and we are confident in our skills to get this done. We also know this is a perfectly affordable project, especially considering that we own many of the components needed.

Bibliography

Cuellar, Joey (MrWiz) “Evo 2016 Numbers - SFV 5065, Smash 4 - 2637, Melee - 2350, Pokken - 1165, GG:XrdR - 903, UMvC3 - 770, MKX - 707, T7 - 543, KI - 540 #Evo2016”. 1 Jul 2016, 12:08 AM. Tweet.

"Evolution Championship Series News and Updates." *ESPN*. N.p., 21 July 2016. Web. 11 Dec. 2016.

“EXI:USBHostAdapter.” *gc-linux.org*.

Ashen. "GameCube Memory Card & SD Gecko Diagrams." *The Official ModRetro Forums*. N.p., 6 Aug. 2011. Web. 11 Dec. 2016.

Gordon, Justin "Adaptive Trigger" "Why Super Smash Bros. Melee HD (with Online) Would Be a Huge Success for Nintendo." *EventHubs*. N.p., 23 Oct. 2016. Web. 11 Dec. 2016.

Groot @AStrayClay, Justin. "EVO 2016 Might Be the Most Competitive Smash Tournament Ever - Kill Screen." *Kill Screen*. N.p., 18 July 2016. Web. 11 Dec. 2016.

“Hardware Enhanced Melee - Proof of Concept.” *YouTube*, uploaded by Fizzi, 30 Aug 2015, “https://www.youtube.com/watch?v=-W9Sczqtik8”

Laferriere, Jas. "Statistics In Melee - The Tech (Part 1)." *Melee It On Me*. N.p., 29 Sept. 2015. Web. 11 Dec. 2016.

Laferriere, Jas. "Statistics in Melee - The Stats (Part 2)." *Melee It On Me*. N.p., 9 Oct. 2015. Web. 11 Dec. 2016.

"LOL News." *TheScore Esports*. N.p., n.d. Web. 11 Dec. 2016.

"VODS." *LoL Esports*. N.p., n.d. Web. 11 Dec. 2016.