Thesis Project Portfolio

Independent Quality Assurance: Letting Interns Test Software with Minimal Instructions

(Technical Report)

Lootboxes in Videogames When They Work and When They do not

(STS Research Paper)

An Undergraduate Thesis

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Michael Rasmussen

Spring, 2022

Department of Computer Science

Table of Contents

Sociotechnical Synthesis

Currently, my STS research project is on the ethics of gambling when used as a payment model in monetizing video games. In that project, I conclude that using what is essentially gambling to make money off of a user base where it is impossible to determine the real age of a given user is unethical on the grounds that it is illegal to offer gambling services to minors. My technical project is a report on the internship I did with the software development company SemanticBits in 2020. During my internship I performed quality assurance testing for a web application the company was developing that would help them with asset management as it was no longer practical for them to keep doing it manually. The outcome of that project was that I got to build some workplace experience and that I found for the app developers several critical bugs that they told me they had not thought to even look for yet. You could say that the relationship between the two projects is that an inventory management app would be helpful to a player of one of many games with scarce resources and the ability to pay money for randomly selected items in the hope that what they want is among them. Such an app was actually my original technical project.

Two years in the making, the subjects of my thesis portfolio have changed quite a lot since I first began the capstone process in the fall of 2019. They have been through so many changes and it has been so long since I started that I forgot what my original topic was about until just now. I thought my original STS research project had been on the history of video game payment models. Imagine my surprise when I looked at my prospectus and saw “Analysis of the History of YouTube Let’s Plays” as my title. I had to double-check that the document that I currently have is the correct one. Now I remember that my STS research was originally about the sociotechnical climate that allowed uploading videogame footage to become a full-time job for

some people. In between semesters I had satisfied my curiosity about my original topic and could write about the actor network of YouTube, YouTubers, Google, alternative video hosting services, etc., and how new developments in video streaming technology, livestreaming, video monetization and demonetization, and changes to personalized content suggestion algorithm all influenced the network. Although I could write a paper on the subject at the time, I found that most of my information on the subject came from my firsthand experience and I couldn't find enough quality sources on the subject to reference for the paper. I almost had the same problem with my current topic on lootboxes. Only after I started my project did a lot of research about lootboxes come out. So one of the things I learned is how it can be difficult to find research about emerging technologies and phenomena as they are happening.

My technical project was only an idea back then, a proposal, submitted in the last week of the semester to a now-retired professor who kindly stood in as my technical advisor because I did not have one; thanks, Professor Alfred Weaver. My technical project began life as a plan to develop an app for players of the game *Warframe* to help optimize their equipment while taking into account the resources available to them. It ended up taking form as an app for the game *Genshin Impact* with the same purpose; where it now lies, half-finished. Due to some unfortunate circumstances I was not able to complete that semester and had to try again later.

The point of my long anecdote is that my current projects, and I myself with them, are both built on the backs of past work; whether in success or failure, completed or otherwise. That is the real relationship between my projects.

Independent Quality Assurance: Letting Interns Test Software with Minimal Instructions

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Michael Rasmussen

Spring, 2022

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Rosanne Vrugtman, Department of Computer Science

Independent Quality Assurance: Letting Interns Test Software with Minimal Instructions

CS 4991 Capstone Report, 2022

Michael Rasmussen
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
mjr8fh@virginia.edu

Abstract

SemanticBits, a software development company headquartered in the Washington, DC, area, needed a better way to manage assets after tripling in size in only two years. SemanticBits decided to develop an app in-house to streamline their asset management process as keeping track of it manually was no longer feasible. I was hired as an intern to perform quality assurance testing for the software. I was given no instructions on what to test for and told to test as I saw fit and ask any questions as they arose. I used the skills I learned throughout my CS coursework to test the product on my own. The bugs and unexpected behaviors I found during my testing proved to be of use and I was able to share my findings and suggest improvements with the developers during regular meetings with them. By not giving me rigid instructions, the company was able to find some important bugs without having to invest very much time or resources into supervising me and investing very little time and resources on me. At the same time, I was provided with an opportunity to build workplace experience in a low-stress environment amidst the COVID pandemic. In the future, I would recommend that the company supply the intern with a definition of the application's intended behaviors and a list of exactly what is ready to be tested so that the intern is not left guessing if a quirky behavior deviates from the app's expected behavior or if an error is the result of a bug or merely the feature not being fully implemented yet.

1 Introduction

SemanticBits is an Indian-American company headquartered in Washington, DC, with offices in Virginia and India. Two years before my internship, the company had less than a hundred employees. By the time of my internship in 2020 the company had grown to 290 employees. Most of their employees work remotely and with their recent growth it was no longer practical to manage the company's assets and projects manually. The company decided to develop a web application in-house to assist them in keeping track of their assets. They called it "eBinja, your business ninja."

With the application, employees could fill and submit requests to check out equipment, confirm delivery of the equipment, submit support tickets if needed, return equipment for maintenance, and return equipment after they finished using it.

The application also allows members of the IT department with administrative accounts to view all of the projects the company has either in-progress or completed and the equipment and personnel assigned to each project. Users could search for specific projects with a search field, or sort projects either by name or project start date. In another tab, users could view all of the equipment in the company's possession. Users could filter assets by status, (in use, in storage, in transit, damaged, or in repair) and sort them by either the asset ID, acquisition date, the name of the project the equipment was checked out for, who the equipment was assigned to or by most recently updated asset. One could also view an asset's assignment history, acquisition

costa and any current/previous shipping information.

2 Related Works

While all courses in the CS program helped prepare me for QA testing, as I always have to test my own code there are some standout courses that were really helpful for my specific internship. In CS 3240 Advanced Software Development [1] I learned about ways to communicate with the client or employer and figure out what they really wanted from me and to help them figure out what it is that they really want themselves. That helped me ask about what they really wanted me to look for since they did not give me any direct instructions. Advanced Software Development also taught me how to make a webapp that uses an SQLite database, knowing how to make one let me understand some common bugs I ran into that I should look for.

Along with Advanced Software Development, there was CS 4720 Mobile App Development [2] and CS 4730 Game Design [3] that all had group projects in them that helped prepare me to work in a team.

CS 3501 Intro to Cybersecurity [4] and CS 4630 Defense Against the Dark Arts [5] both taught me about some vulnerabilities and exploits that I could test for that I would not have known how to test for otherwise. And of course, there was CS 3250 Software Testing [6] that taught me how to make better tests and expand my testing coverage.

3 Process Design

After my mentor gave me a brief overview of what the app does and gave me admin login credentials, they told me to just test as I saw fit. The first thing I asked about was if there was anything in particular they wanted me to test or did not need to test and they said that I did not need to test user creation and account management. The second thing I had to ask was how badly they wanted me to try to break the software, because if the app failed the basic cybersecurity tests I was going to perform, the database could get wiped.

After confirming the scope of what they wanted me to test and that there was nothing in particular that they wanted me to test for, I tried to break the system, with permission granted, by performing a very basic SQL injection attack. I looked through the HTML data of the app for anything exploitable in a cross-site scripting attack. The tests passed, as they should have, since what I practiced in my cybersecurity classes is supposed to be automatically protected against by the latest programming languages.

I then started making tests for the functionality of the application itself. I did not have a way to automate the process, so I had to input all of the data into the form fields manually. I started out testing each interactable component of the app (mostly buttons and drop-down lists) individually to see if they worked on their own when given expected user input.

Next, I tested each component while supplying an invalid input to make sure that the app did not crash and was able to process bad requests. Then I tested all of the components together while supplying the app with the valid inputs that a typical user might make while using the app. So, in this case, it was uploading a test document for an invoice for purchasing new equipment; filling out the forms and uploading the documentation for receipt of the equipment; and filling out the forms for checking the equipment in. Then I would check out the equipment to a user, upload a test document for a shipping invoice and upload documentation confirming the recipient received the equipment and its current condition. I would repeat variations of the whole process while changing input variables to try to best account for what a user should do, would do, or could do, and try to test edge cases in the input domain.

I maintained a log with reproducible steps of all of the tests I ran to keep track of my battery of tests to repeat later and in case anyone wanted to see what I was doing or wanted a written log of steps. If the app failed any test, I would then go a step further and figure out what specifically was causing the problem to make it easier for the developers to fix. So instead of saying, “the app broke when I supplied it with these inputs,” I

would say, “the app will break anytime this specific field or fields contain variables belonging to this set of values.”

During my investigations into what was causing the bug, if I found more information, I would strike out the original description of the bug and update it to include new findings so I could visually see the progress I had made over time.

At the end of the day, if I found any bugs or had suggestions for improvements, I would meet with someone from the development team in India over a video conference and discuss what I found with them. I would tell them about the bugs or strange behaviors I found and show them how to reliably reproduce it. I would also inform them of behaviors I observed that I knew could easily cause problems down the road, but did not know whether or not what I found deviated from the expected behavior of the app.

The next day, if they patched the application to address the issues I reported, I would test to see that the bug was indeed fixed. Then I would repeat every test I had previously performed to make sure that everything that used to work still functioned as intended. Sometimes it did not. After I finished verifying that nothing had broken since the last build and whether or not the bugs were patched, or if they had yet put out an update yet, I would begin my testing process anew. I would try to think of new edge cases, unexplored areas of the input domain I had yet to test, and complications that could arise from the implementation of the app that were not due to programming errors.

4 Results

During my short time testing eBinja, I managed to identify quite a few errors. The copy of my testing documentation that I maintained on my personal computer is missing a few pages that I kept on a notepad at the office before I went digital, but it is still ten pages long. I tried my best to sort the issues I found into a few categories: “major bugs” which render the application unusable, “minor bugs” that impair the application but do not disable it, “trivial bugs” that are mostly cosmetic in nature and do not affect the application’s usability, and “strange behaviors” which were behaviors of the application that I either could not tell whether they were the outcome intended by the developers or behaviors that I knew would eventually cause a problem unless the user of the app was explicitly aware of the issue and took measures to avoid it.

I mentioned in my process design that I would strike through old information if I found some insight into a bug’s true nature that I initially did not recognize or needed more testing to confirm. For example, one entry previously stated “the ‘Action Date’ field in editable information submission forms defaults to either the current date, the delivery date on the contract page, or the current system date. All dates were the same in the test and more testing is required.” That was stricken through, but left in the document to chronicle my testing progress. Above it is a more succinct “Action Date field defaults to system clock date,” while below it is a note detailing the significance of what I found.

The significance was that, while defaulting to the current time on the device’s system clock is great for the asset creation and asset action forms, the same cannot be said for the “edit asset information” and “edit asset history” pages which use the same form, but preload all fields, besides the date, with information on the asset stored in the database. It was a problem that is easy to work around if one knew it was there, but also easy for a user to miss. The wrong date would be saved to the database eventually if the user was not aware of this quirk or had forgotten to manually re-enter the original date.

That is just one detailed example of some of the things I found. Other bugs I found included:

- The last four digits of credit card information not saving if there were leading zeroes
- A form field was not marked as required when it actually was and the user would not be told what was preventing a contract from saving
- The “ComplaintRegistered” action would not update the “last action taken” field

- When submitting a repair form, asset condition would not change to “damaged on arrival” or “damaged” accordingly
- Billing info would only display the original information saved and would not reflect any future edits
- Comments over a certain size throw a generic “error 400” without telling the user what they did wrong or preventing the user from typing too many characters in the first place
- Comment history would display oldest to newest requiring the user to scroll ever increasingly downward just to look at the most recent comment; clicking “save” on the edit form sets the last action date to whatever is in the delivery date field
- Drop down menus have fillable text boxes that filter out options from the selection, but if text is left in the search box and nothing from the drop down menu was clicked, an error would occur if that text did not match an entry in the database
- Two specific drop-down menus still have selected text boxes after selecting from the menu after a supposed patch to fix the previous bug
- One specific drop-down menu now gets cut off by another frame in the context
- One field lost its “required field” tag but is still required.

Some of the major bugs I found are worth describing in more detail. I found one that took a long time to diagnose because it involved many variables but the short version is that full payment info would not save at contract creation or on the same edit as the first time a payment method is saved. It will save if no payment method is selected and will remain saved if a payment method is selected later. It will also save and remain saved if the payment method is being changed, as long as it is not the first time a payment method was added.

A major but simple bug I found is that credit card information would not save if the last four digits of the card number had leading zeroes.

I classed this next one as a minor bug but the developers told me it was more important than I thought. The ordering of assets on a multi-asset contract was not preserved if you try and fail to save because you did not assign an owner or disposition for each asset on your first attempt at saving, even if you correct the form and submit the save request successfully the next time.

Another bug that I classed as minor (due to my presumed complexity of fixing the bug) that turned out to be more important was that the “invoice paid” checkbox would never save on contract creation. You would always need to check it later, which would typically happen when you upload the invoice.

On one of the builds of the app during my routine tests I ran whenever a new build was released to validate that features that used to work still function, I noticed immediately that the billing info page for contracts would no longer display multiple assets or unit price, but still displayed the sum total of their prices. The CSV table that the app exported still displayed unit price, but only for the first asset in the contract. I do not know how they missed it, but they confirmed it was a bug they missed.

5 Conclusion

During my internship I was able to get experience in a professional work environment. Granted, the office only had three people in it including myself because everyone else was working from home due to the COVID-19 pandemic. I had the option to work at home too, but I wanted some experience at an office. I was given free rein to test however I wanted. I learned that I was better at testing than I thought based on my performance. They told me that I was testing for and finding bugs that they had not even thought to look for yet. I feel like I contributed significantly to their project with my detailed reports and suggestions. I would say that the project was a success. I got the work experience I wanted and they got free testing for around 30 hours a week.

6 Future Work

Something my internship could improve on would be to better communicate the current state of the project's development and/or provide the tester with a list of known problems.

After using the app, my first impression was that some features were still in development and not ready for testing. It was only after I asked what was ready to be tested, and after being praised for finding bugs that I thought that, surely, they already knew about, did I report on a lot of bugs I had found but did not want to bother them with because I thought they were known issues.

References

[1] Paul McBurney. 2022. Advanced Software Development. Retrieved April 24, 2022 from http://cs3240.cs.virginia.edu/

[2] Mark Sherriff. 2018. Mobile Application Development. Retrieved April 24, 2022 from http://cs4720.cs.virginia.edu/f18/

[3] Mark Floryan. 2020. Game Design. Retrieved April 24, 2022 from https://markfloryan.github.io/gamedesign/readme.html

[4] Aaron Bloomfield. 2019. Introduction to Cybersecurity. Retrieved April 24. 2022 from https://aaronbloomfield.github.io/ics/readme.html

[5] Aaron Bloomfield. 2017. Defense Against the Dark Arts. Retrieved April 24. 2022 from https://aaronbloomfield.github.io/dada/readme.html

[6] Upsorn Praphamontripong. 2022. Software Testing. Retrieved April 24, 2022 from https://www.cs.virginia.edu/~up3f/cs3250/syllabus.html

LOOTBOXES IN VIDEOGAMES: WHEN THEY WORK AND WHEN THEY DO NOT

A Research Paper submitted to the Department of Engineering and Society
In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science in Computer Science

By

Michael Rasmussen

May 3, 2022

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

ADVISOR
Hannah S. Rogers, Department of Engineering and Society

LOOTBOXES IN VIDEOGAMES: WHEN THEY WORK AND WHEN THEY DO NOT

THE COST OF GAME DEVELOPMENT HAS BEEN RISING WHILE THE PRICE OF GAMES HAS NOT

Video games are bigger than ever. With 2.5 billion gamers around the world spending $152.1 billion on games in 2019 (Wijman, 2019), money generated from video games surpassed the revenues earned by the movie and music industries combined. For comparison, in 2019 the global box office was worth $41.7 billion and revenue from the music industry reached $19.1 billion in 2018 (Stewart, 2019). Video games are bigger than ever in a more literal sense too. Both the size in bytes and development cost, adjusted for inflation, have gone up tenfold every ten years since 1995 (Koster, 2018). Despite this, the price of a AAA game has not risen since 2005, when the price of a game rose from $50 to $60 for the 7th generation of consoles (Witkowski, 2020). To combat this, developers have had to come up with new ways to either fund development or monetize their games.

Video game developers have come up with a number of ways to make more money off of a game after a user's initial purchase of the game. These include monthly subscriptions, new purchasable, downloadable content (DLC), in-game purchases (microtransactions), and most recently, a shift towards a "games as a service" type model characterized by the widespread adoption of the battle pass. For the scope of this paper, I will be looking at lootboxes, a microtransaction where users pay for a random assortment of items drawn from a known pool of items.

The STS research paper will use Wiebe Bijker and Trevor Pinch's (1984) social construction of technology (SCOT) theory as a method to analyze the performance of lootboxes

as a technology to address the problem of the rising costs of videogame development. The paper will use two games as case studies to show how the interpretative flexibility of the lootbox can lead to either a massive disaster or a huge success because the design flexibility of the lootbox allows it to lead to different implementations and different social groups respond differentlyto those implementations.

THE RISING COST OF DEVELOPMENT

As video games are becoming more complex over time with the development of more advanced computers, so too has the development cost increased alongside it. In his study analyzing more than 250 games, Raph Koster (2018) demonstrated that the development cost of games has increased roughly tenfold every ten years. This can be seen in Figure 1, Koster's plot of the cost of games adjusted for inflation for 2017. Keep in mind that only the development costs of games (including salaries) were recorded and that marketing budgets of AAA games

tend to be from 75% to 100% as much as the cost of development (Koster, 2018).

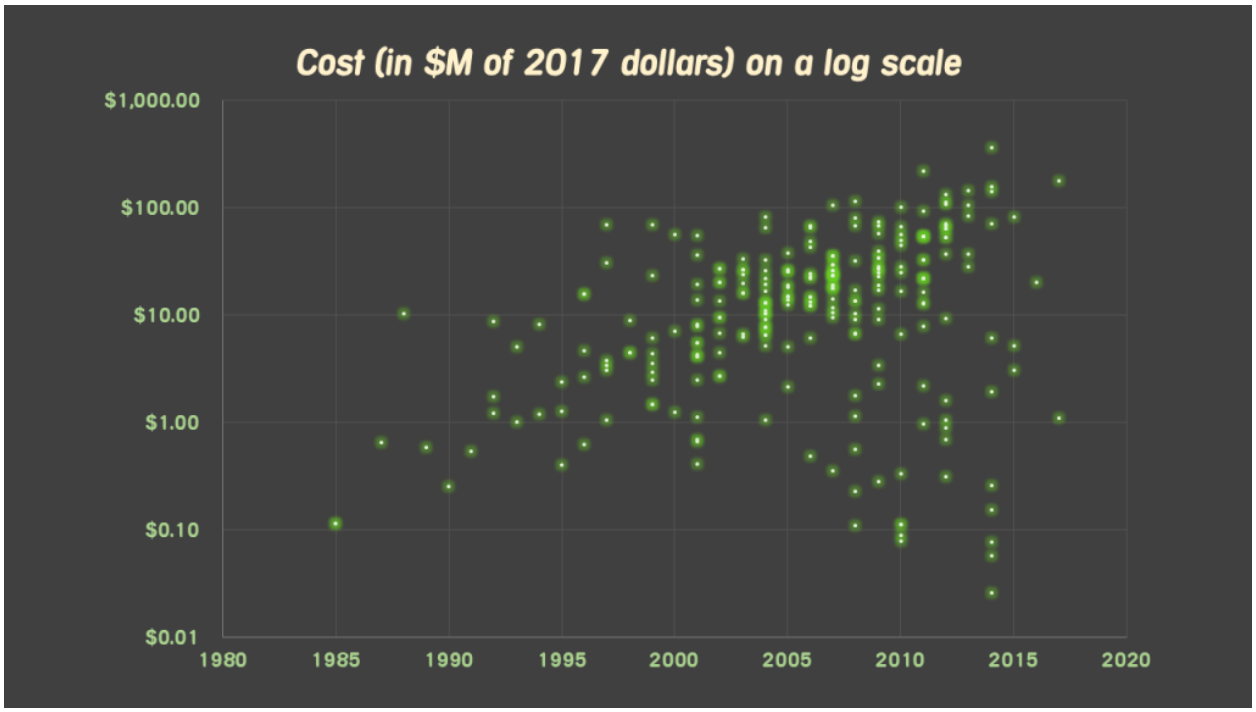

Figure 1: Cost to Develop Games in 2017 Dollars. This figure shows costs in 2017 dollars to develop video games. Koster notes that the cost of game development is likely to be over-reported as huge budgets make for more interesting reporting. (Adapted by Michael Rasmussen (2021) from Raph Koster 2018).

You can see that the average cost went up ten times every ten years. You can also see small budget indie hits on the graph and cheaper games being made after 2005 when the game engines Unreal Engine 3 and Unity came out. Similarly, Figure 2 shows that the number of bytes per game also went up at least ten times every ten years. It is interesting that the number of bytes per game didn't increase stepwise with each console generation. Koster suggests that this may be because of advances in data compression and run-time decompression.

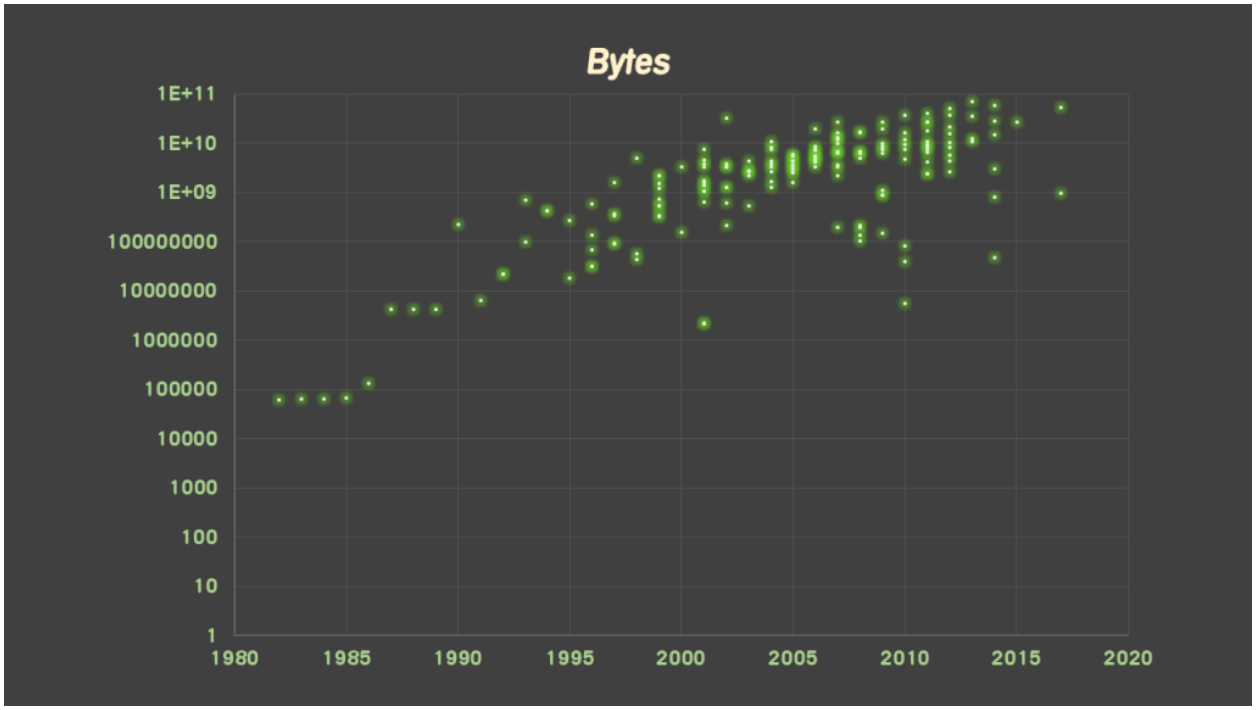

Figure 2: Number of Bytes per Game on a Logarithmic Scale. The number of bytes per game dramatically went up over the years. Koster notes game size is likely to be underreported due to advancements in data streaming. (Adapted by Michael Rasmussen (2021) from Raph Koster 2018).

Knowing the cost to develop games and the number of bytes per game, it is possible to plot the costs per byte. Figure 3 shows that there is about 10 times variability in cost within a given year. Koster says that most of the variability can be explained by whether a game is content driven or system driven. For example, an expansive world with a ton of assets will cost more to make than maps for a competitive multiplayer game.

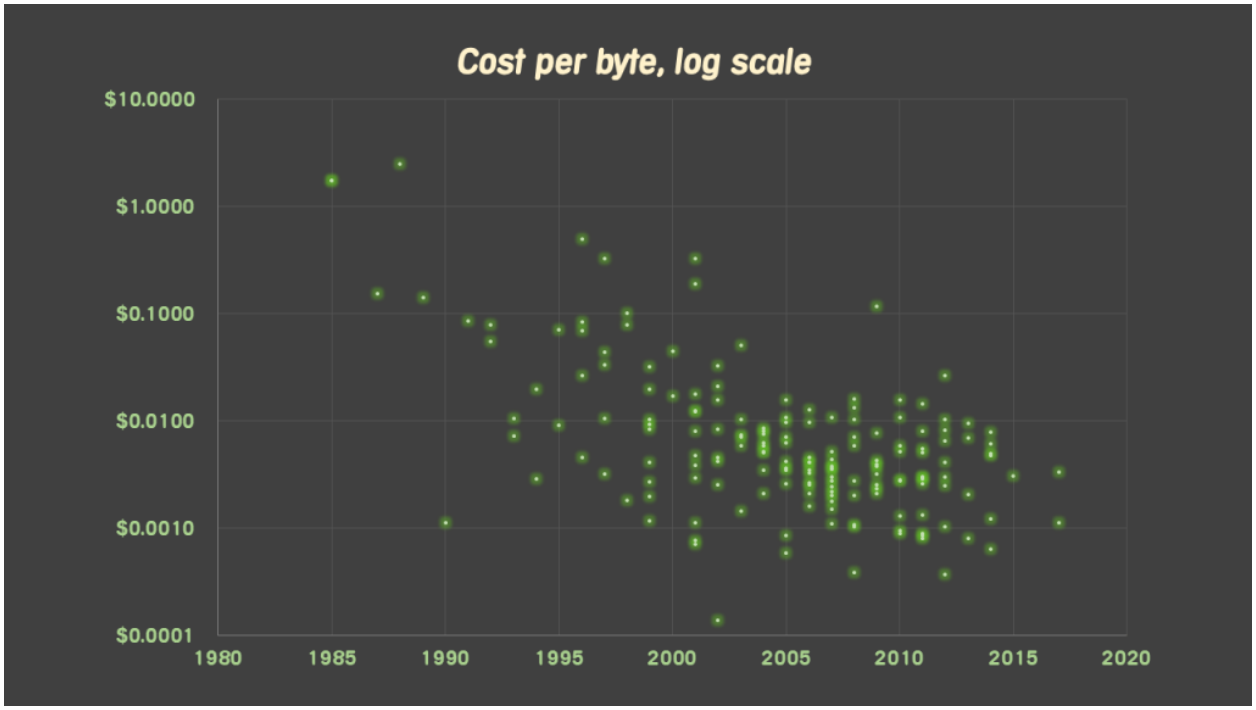

Figure 3: Cost per Byte on a Logarithmic Scale. There is about ten times variability in cost in a given year (Adapted by Michael Rasmussen (2021) from Raph Koster 2018).

Looking at the graph, it looks like the average cost per byte has plateaued. Unreal Engine 3 and Unity both launched in the 2004 to 2005 time window and rather than helping the cost per byte, it appears to have flattened the curve. Koster raises the possibility that perhaps standardizing on the two engines blocked further innovations on techniques that reduce costs. Either way, video games are getting bigger and bigger so the total cost of development is increasing over time.

Social Groups Relevant to Lootboxes

With the size of games ever-increasing and the cost per byte having plateaued since 2005, video game publishers and developers have had to find more ways to make money off of their games if they wanted to keep the base sale price at $60. A method of monetization that has been gaining traction for a long time is the “lootbox” model, wherein players are incentivized to spend

real money in exchange for random items from a given pool, with the most desired items often also being the least likely to be selected.

At its most basic level, the relevant social groups for lootboxes are the users playing the game and the developers making the game. However, many subgroups can be identified. Influencing the developers besides the players are publishers, intellectual property owners if the game is a licensed game, shareholders if applicable, and other games which they draw inspiration from. The players are influenced by game critics, experiences from other games, and the game itself.

An interesting social group is that of lawmakers. In the United States there are no laws directly governing what is allowed to be in a videogame. Following guidelines under the Entertainment Software Ratings Board (ESRB) is a self-regulatory process. Most retailers will not stock unrated games or games with an adults only (AO) rating, so developers that are not distributing their products exclusively through digital marketplaces are inclined to opt in to the rating system. For a good example, the infamous “Hot Coffee” scandal of Rockstar Game’s Grand Theft Auto: San Andeas, where unused code for a sex minigame was found still on the disc and modded in, was not inherently illegal for being sexually explicit, but was declared illegal in that the developers engaged in false advertising by changing the rating of the game to AO by not originally disclosing the content on the disc when an 85-year old woman filed a class action lawsuit against Rockstar and parent company Take-Two Interactive (Render, 2021). However, developers need to comply with each countries’ laws before their games can be sold there. This is actually why a bunch of games with loot boxes in them disclose their drop rates now, because they are required to do so in China (Gartenberg, 2017). They only have to disclose that information in the Chinese version, but most developers disclose the information in all

versions of their games now. On the other hand some developers decide that it is just not worth the trouble to comply with local laws and will just pull features or not sell the game there as many did Belgium after they declared loot boxes illegal (Gerken, 2018).

Design Flexibility and Interpretations

Lootboxes can be implemented in different ways. The two biggest decicions are whether or not the contents of the lootbox are cosmetic only, or offer gameplay effecting items, and whether or not players have to buy the lootboxes or if they can earn them by playing the game and if so, if there is a limit on how many they can earn.

Full priced games with DLC and lootboxes are often heavily criticized by players especially if the game is singleplayer only or will not be receiving new content. Developers and Publishers consider loot boxes as one way to generate revenue post-release to avoid raising the cost of a game past $60. Many players consider loot boxes and microtransactions justified if it helps the developers deliver free content updates or especially if the game was already free-to-play.

The decision to include loot boxes in a given game may come from the developers of the game or from their publishers, but the decisions of their implementation are often set by the developers themselves (Fenlon, 2017).

An Example, Star Wars: Battlefront II

Video game publisher and developer Electronic Arts' (EA) game *Star Wars: Battlefront II* is a great example of the various social groups at play in influencing the contents of a game. In the case of Battlefront II, weeks before the game launched, EA showcased that players were to

acquire characters and upgrades randomly through loot boxes (Park, 2017). The public backlash resulted in EA trying to defend itself on Reddit, where they still hold the most downvoted comment in Reddit history, at -683,000 (EACommunityTeam, 2017). For comparison, the next most disliked comment had a score of -88,900 and was a deliberate attempt to see how many dislikes it could garner (IranianGenius, 2021). The negative response from players and media coverage prompted Disney's (the intelectually property rights holder) head of interactive media to send a message to EA expressing concerns about how the outcry reflected negatively on Disney's property (Fritz & Needleman, 2017). Belgium's gaming commission investigated whether loot boxes in Battlefront constituted gambling, putting EA in the legal spotlight (Moon, 2017). In a rare case of developers interacting with shareholders, EA felt the need to file a note with the Securities and Exchange Commission stating that they didn't expect their decision to remove loot boxes to have a significant impact on their fiscal year (Schatz, 2017).

With Battlefront II the problem wasn't the loot boxes themselves, outside of countries that ban games with loot boxes in them at least, loot boxes had been established as a gaming monetization model for a while by the time of the game's release. The problem was what was inside the loot boxes. Most players agree that loot boxes are fine as long as the contents offer no gameplay advantage and are purely cosmetic. Less so are they okay with them containing contents that offer a competitive advantage. This was to be the case with Battlefront II. Not only were playable character unlocks locked behind loot boxes, but Battlefront II's player progression system was built on its loot boxes. Instead of the more traditional level-based system where players level up and are given upgrades at specific levels, upgrades too were to be acquired from loot boxes earned from gameplay or purchased with real money.

Another Example: Genshin Impact

But at least you can earn unlimited loot boxes in games like Battlefront II if you played the game enough. Not so is the case with gacha games. Gacha games are typically free-to-play mobile games with the gacha, basically a Japenese lootbox, being the main way to obtain new content in-game. The name comes from the gachapon toy vending machines that dispense random capsule toys when you insert a coin or coins and turn a crank. In a typical free-to-play gacha game, your gameplay is limited by an energy system that recharges in real-time with players having the option to pay money to recharge it instantly. Characters and/or equipment are obtained from the gacha mechanic, which is essentially the same thing as the loot box system. In a typical gacha game, only a limited amount of in-game currency is earnable by the player at any given time and since the player only has a random chance of getting whatever item they want from the gacha it is often likely that they won't be able to obtain what they want with they can earn by playing the game. The way they make so much money lies in that to obtain a particular item from the gacha, a player would often have to spend more than the $60 price of a retailing AAA title. Again it is random what the player receives for the money they spent and they could possibly never end up with what they want. As an example of how powerful the loot box or gacha system can be, the mobile version of the free-to-play game Genshin Impact, the world's first AAA gacha game, made $874 million in its first five months becoming the third best-selling mobile game ever (Chapple, 2021). The game is also playable on Playstation and Windows but only revenue generated from the mobile version of the game is factored into the previous statistic.

It is worth noting that gacha games are very popular in the East, hence why it has its own name to separate it as an entire genre compared to games that also have loot boxes in them. Also

of note is that, to bring back the terms of social groups and interpretatitive flexibility, non-Western gamers and younger gamers are more accepting of lootboxes in games because that is “just the way it is”, it was what they grew up with (Brightman, 2017).

Closure

Following the controversy of Battlefront II developers and publishers have pulled loot boxes from their games. Existing games such as Star Wars Battlefront II, Dauntless, Middle-Earth: Shadows of War, Forza Motorsport 7, and Rocket League have removed their lootboxes and new games are moving on to the battle pass method such as Call of Duty, Gears of War, and Halo where the previous entry in the series (Call of Duty: Black Ops 4, Gears of War 4, and Halo 5) had loot boxes but the next installment (Call of Duty: Modern Warfare, Gears 5, Halo Infinite) all have battle passes instead. However, gacha games are still going strong in the East and Genshin Impact has shown they work just as well in the west with Genshin Impact making $3 billion USD within 18 months of its release (Murray, 2022).

THE FUTURE

In the pursuit of profits, companies are often looking to make the most money they can off of the least effort possible if they are allowed to do so. It is worth noting that while many games have stopped using loot boxes in favor of the battle pass model, just as many have added the battle pass model on top of the loot box model. The cost of development will continue to increase in the next console generation and the new price increase to $70 (Kharf & Mochizuki, 2020) will probably not be enough to cover it.

REFERENCES

Brightman, J (2017, November 20). Loot boxes are not bad game design, say devs. *Gameindustry.biz.* https://www.gamesindustry.biz/articles/2017-11-20-loot-boxes-are-not-bad-game-design-say-devs

Chapple, C (2021, March). Genshin Impact hits $874 million spent since launch, already the world's third highest earning mobile game. *Sensor Tower Blog*. https://sensortower.com/blog/genshin-impact-revenue-first-five-months

EAComunityTeam. (2017). r/StarWarsBattlefront - comment by u/EACommunityTeam on "seriously? I paid 80$ to have Vader locked?". *Reddit*. https://www.reddit.com/r/StarWarsBattlefront/comments/7cff0b/seriously_i_paid_80_to_have_vader_locked/dppum98

Fenlon, W. (2017, November 21). The case for and against loot boxes, according to developers. *PC Gamer.* https://www.pcgamer.com/the-case-for-and-against-loot-boxes-according-to-developers/

Fritz, B., & Needleman, S. (2017, November 17). Electronic arts pulls microtransactions from 'Star Wars Battlefront II' after fan backlash. *Wall Street Journal*. https://www.wsj.com/articles/electronic-arts-pulls-microtransactions-from-star-wars-battlefront-ii-after-fan-backlash-1510936871

Gartenberg, C. (2017, May 2). ChinaChina's new law forces Dota, League of Legends, and other games to reveal odds of scoring good loot. *The Verge.* https://www.theverge.com/2017/5/2/15517962/china-new-law-dota-league-of-legends-odds-loot-box-random

Gerken, T. (2018, April 26). Video game loot boxes declared illegal under Belgium gambling laws. *BBC News*. https://www.bbc.com/news/technology-43906306

IranianGenius. (2021). downvoted—ListOfComments. *Reddit*. https://www.reddit.com/r/ListOfComments/wiki/downvoted

Kharf, O., & Mochizuki, T. (2020, November 9). Video game prices are going up for the first time in 15 years. *Bloomberg*. https://www.bloomberg.com/news/articles/2020-11-09/game-prices-go-up-to-70-the-first-increase-in-15-years

Koster, R. (2018, January 24). The cost of games. *VentureBeat*. https://venturebeat.com/2018/01/23/the-cost-of-games/

Moon, M. (2017, November 16). "Battlefront II" under investigation in Belgium over loot crates. *Engadget*. https://www.engadget.com/2017-11-16-battlefront-ii-belgium-gambling-investigation.html

Murray, S. (2022, May 05). Genshin Impact has made $3 billion in less than two years. *TheGamer*. https://www.thegamer.com/genshin-impact-3-billion/

Park, G. (2017, November 18). How a Star Wars video game faced charges that it was promoting gambling. *Washington Post*. https://www.washingtonpost.com/news/comic-riffs/wp/2017/11/18/how-a-star-wars-video-game-faced-charges-that-it-was-promoting-gambling/

Pinch, T. J., & Bijker, W. E. (1984). The social construction of facts and artefacts: or How the sociology of science and the sociology of technology might benefit each other. *Social studies of science*, *14*(3), 399-441. https://doi.org/10.1177/030631284014003004

Render, J (2021, October 26). What was Grand Theft Auto: San Andreas' hot coffee controversy? *CBR*. https://www.cbr.com/what-was-grand-theft-auto-san-andreas-hot-coffee-controversy/

Schatz, J. (2017). *Current report pursuant to Section 13 or 15(d) of The Securities Exchange Act of 1934*. https://www.sec.gov/Archives/edgar/data/712515/000071251517000095/form8kfiled20171117.htm

Stewart, S. (2019, October 22). Video game industry silently taking over entertainment world. *EJINSIGHT*. https://ejinsight.com/eji/article/id/2280405/20191022-video-game-industry-silently-taking-over-entertainment-world

Wijman, T. (2019, June 18). The global games market will generate $152.1 billion in 2019 as the U.S. overtakes China as the biggest market. *Newzoo*. https://newzoo.com/insights/articles/the-global-games-market-will-generate-152-1-billion-in-2019-as-the-u-s-overtakes-china-as-the-biggest-market/

Witkowski, W. (2020, October 9). Videogames are about to get more expensive for the first time in 15 years. *MarketWatch*. https://www.marketwatch.com/story/the-age-of-70-videogames-has-arrived-11602257572

BIBLIOGRPAHY

Agarwal, P. (2017, November 19). Microtransactions in video games. *Intelligent Economist*. https://www.intelligenteconomist.com/microtransactions/

Ball, C., & Fordham, J. (2018). *Monetization is the message: A historical examination of video game microtransactions*. 4.

Baritaud, C. (2021). Summary of some analytical approaches. [Class handout].

Bryan, W. (2021, February 2). How much does it cost to make a video game? *The Ultimate Resource for Video Game Design*. https://www.gamedesigning.org/learn/video-game-cost/

Chapple, C. (2021, March 4). Genshin Impact hits $874 million spent since launch, already the world's third highest earning mobile game. *Sensor Tower Blog*. https://sensortower.com/blog/genshin-impact-revenue-first-five-months

Dingman, H. (2017a, October 10). How loot boxes are turning full-priced PC games into pay-to-win games of chance. *PCWorld*. https://www.pcworld.com/article/3231668/loot-boxes-ruining-gaming.html

Dingman, H. (2017b, November 10). Need for Speed: Payback review: Pouring loot boxes on a tire fire. *PCWorld*. https://www.pcworld.com/article/3236057/need-for-speed-payback-review.html

Dingman, H. (2020, January 17). Battle passes are making me play games less, not more. *PCWorld*. https://www.pcworld.com/article/3513298/battle-passes-are-making-me-play-games-less-not-more.html

EAComunityTeam. (2017). *R/starwarsbattlefront - comment by u/EAcommunityteam on "seriously? I paid 80$ to have vader locked?"*. Reddit. https://www.reddit.com/r/StarWarsBattlefront/comments/7cff0b/seriously_i_paid_80_to_have_vader_locked/dppum98

Fritz, B., & Needleman, S. (2017, November 17). Electronic Arts pulls microtransactions from 'Star Wars Battlefront II' after fan backlash. *Wall Street Journal*. https://www.wsj.com/articles/electronic-arts-pulls-microtransactions-from-star-wars-battlefront-ii-after-fan-backlash-1510936871

Hume, M., Klimentov, M., Favis, E., Park, G., & Amenabar, T. (2020, December 30.). Analysis | The biggest questions facing the gaming world in 2021. *Washington Post*. Retrieved March 4, 2021, from https://www.washingtonpost.com/video-games/2020/12/30/2021-video-game-outlook/

IranianGenius. (2021). *downvoted—ListOfComments*. Reddit. https://www.reddit.com/r/ListOfComments/wiki/downvoted

Johnson, M. R., & Brock, T. (2020). The “gambling turn” in digital game monetization. *Journal of Gaming & Virtual Worlds*, *12*(2). Computers & Applied Sciences Complete. https://doi.org/10.1386/jgvw_00011_1

Joseph, D. (2020). Battle pass capitalism. *Journal of Consumer Culture*. https://journals.sagepub.com/home/joc

Kerslake, W. (2016, October 15). *Why have video game budgets skyrocketed in recent years? - Quora*. Quora. https://www.quora.com/Why-have-video-game-budgets-skyrocketed-in-recent-years

Kharf, O., & Mochizuki, T. (2020, November 9). Video game prices are going up for the first time in 15 years. *Bloomberg*. https://www.bloomberg.com/news/articles/2020-11-09/game-prices-go-up-to-70-the-first-increase-in-15-years

King, D. L., Delfabbro, P. H., Gainsbury, S. M., Dreier, M., Greer, N., & Billieux, J. (2019). Unfair play? Video games as exploitative monetized services: An examination of game patents from a consumer protection perspective. *Computers in Human Behavior*, *101*, 131–143. https://doi.org/10.1016/j.chb.2019.07.017

Koksal, I. (2019, November 8). Video gaming industry & its revenue shift. *Forbes*. https://www.forbes.com/sites/ilkerkoksal/2019/11/08/video-gaming-industry--its-revenue-shift/

Koster, R. (2018, January 24). The cost of games. *VentureBeat*. https://venturebeat.com/2018/01/23/the-cost-of-games/

Madigan, J. (n.d.). What the heck are surprise mechanics? *Forbes*. Retrieved February 18, 2021, from https://www.forbes.com/sites/jamiemadigan/2019/08/19/what-the-heck-are-surprise-mechanics/

Mann, A. (2019). Pseudo-gambling and whaling: How loot boxes prey on vulnerable populations and how to curtail future predatory behavior. *Washington Journal of Law, Technology & Arts*, *15*(3), 200–246. HeinOnline.

Moon, M. (2017, November 16). *“Battlefront II” under investigation in Belgium over loot crates*. Engadget. https://www.engadget.com/2017-11-16-battlefront-ii-belgium-gambling-investigation.html

Park, G. (2017, November 18). How a Star Wars video game faced charges that it was promoting gambling. *Washington Post*. https://www.washingtonpost.com/news/comic-riffs/wp/2017/11/18/how-a-star-wars-video-game-faced-charges-that-it-was-promoting-gambling/

Park, G. (2020a, December 8). With 100 billion hours watched on YouTube for gaming, the site prepares for global growth. *Washington Post*. https://www.washingtonpost.com/video-games/2020/12/08/youtube-gaming-2020/

Park, G. (2020b, December 23). How Xbox learned to sell its consoles like smart phones. *Washington Post*. https://www.washingtonpost.com/video-games/2020/12/23/xbox-series-x-smart-phone-sales-strategy/

Park, G. (2021a, January 27). Perspective | Video games deserve better than blanket, parachute coverage from reporters who don't get it. *Washington Post*. https://www.washingtonpost.com/video-games/2021/01/27/video-games-screen-time-new-york-times/

Park, G. (2021b, February 17). 'Final Fantasy XIV' director discusses the challenges of building an always-online game. *Washington Post*. https://www.washingtonpost.com/video-games/2021/02/17/final-fantasy-xiv-updates/

Petrovskaya, E., & Zendle, D. (2020). *The battle pass: A mixed-methods investigation into a growing type of video game monetisation*. OSF Preprints. https://doi.org/10.31219/osf.io/vnmeq

Richtel, M. (2021, January 16). Children's screen time has soared in the pandemic, alarming parents and researchers. *The New York Times*. https://www.nytimes.com/2021/01/16/health/covid-kids-tech-use.html

Schatz, J. (2017). *Current report pursuant to section 13 or 15(d) of the Securities Exchange Act of 1934*. https://www.sec.gov/Archives/edgar/data/712515/000071251517000095/form8kfiled20171117.htm

Smith, N. (n.d.). Virtual reality is starting to see actual gains in gaming. *Washington Post*. Retrieved March 4, 2021, from https://www.washingtonpost.com/video-games/2021/02/04/virtual-reality-future-games/

Stewart, S. (2019, October 22). Video game industry silently taking over entertainment world. *EJINSIGHT*. https://ejinsight.com/eji/article/id/2280405/20191022-video-game-industry-silently-taking-over-entertainment-world

Tomić, N. (2019). *Economic model of microtransactions in video games*. *1*. https://doi.org/10.30564/jesr.v1i1.439

Wijman, T. (2019, June 18). The global games market will generate $152.1 billion in 2019 as the U.S. overtakes China as the biggest market. *Newzoo*. https://newzoo.com/insights/articles/the-global-games-market-will-generate-152-1-billion-in-2019-as-the-u-s-overtakes-china-as-the-biggest-market/

Witkowski, W. (2020, October 9). Videogames are about to get more expensive for the first time in 15 years. *MarketWatch*. https://www.marketwatch.com/story/the-age-of-70-videogames-has-arrived-11602257572

Zanescu, A., French, M., & Lajeunesse, M. (2020). Betting on dota 2's battle pass: Gamblification and productivity in play. *New Media & Society*, 1461444820941381. https://doi.org/10.1177/1461444820941381

Building a Mobile App with JavaScript to Help
Make and Compare Builds in Warframe
(Technical Paper)

Analysis of the History of YouTube Let's Plays

(STS Paper)

A Thesis Prospectus Submitted to the

Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements of the Degree
Bachelor of Science, School of Engineering

Michael Rasmussen
Fall 2019

Technical Project Team Members
Michael Rasmussen

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Signature

Engineering and Society

_____
Michael Rasmussen

Approved

_____
Alfred Weaver, Department of
Computer Science

Approved

_____
Kent Wayland, Department of

Date_____ Date ____

Date _____

General Research Problem

When will people who don't play videogames stop thinking they are a waste of time? Videogames started life out being made and marketed as entertainment for children. As those children matured, so did the videogames they play, not only in the complexity of the technology, but also in the complexity and maturity of the subject matter. Gone are the arcade games kids used to play for points towards a high score, and in their place are rich worlds full of compelling characters and gripping narratives. A lot has changed about videogames and the industry as a whole in the relatively small amount of time the industry has existed, but what hasn't changed nearly as much is the perception of videogames by those who do not play them.

Videogames are certainly not just for kids anymore, with the majority of top-selling games containing mature subject matter made for adults, but among people who don't play videogames, there are more than double those that think videogames are a waste of time than do not. A study conducted by the Pew Research Center surveying 2000 adults in the United States found that, among adults who don't play video games, 33% said that most games are a waste of time, 29% said that some are and others are not, and only 13% said that most games are not a waste of time; the remaining 25% were uncertain of whether they thought videogames were a waste of time or not (Duggan, 2015).

Building a Mobile App to Help Make and Compare Builds in Warframe

Warframe doesn't currently offer a way to compare mod configurations across different weapons. Warframe is a videogame with nearly 50 million registered players, 12 million of which registered just last year (Bailey, 2019). Even if you think videogames are a

waste of time, an app that lets you make and compare builds for Warframe would be useful for a lot of people. Put simply, in Warframe, you can pick up to eight upgrades to put on a weapon so you have to pick and choose the best ones. The game shows you the modified statistics of the weapon, but you always have to calculate damage per second by yourself. Players have made web apps that will calculate most of the relevant statistics you need as you plan your build on the site, and while it is so much better than working in spreadsheets for those of us that care about optimizing our weapons' damage output, the process for saving builds is a little bit clunky. There is also no way to compare builds across weapons in the app so you still have to do it manually if you want to see a side by side comparison of your alternative weapon options.

I want to use JavaScript and React to make a mobile app for building, saving, and comparing weapon mod configurations to hopefully cut down on the hours spent optimizing your equipment. At the very least, this app will let you save your builds locally, and organize them outside of the game, and you won't need to have an external site or spreadsheet open in another tab while you copy the build you made over to the game.

Analysis of the History of YouTube Let's Plays

How did the current state of Let's Plays on YouTube come to be? Right now, it is possible to make a living playing video games. People can do this by uploading video footage of themselves playing videogames to video hosting services like YouTube or by streaming their gameplay live on platforms like Twitch. Videos such as these are commonly referred to as "Let's Plays." But how did the current state of Let's Plays on YouTube come to be?

Introduction

Videogames have come a long way since the crash of 1983 when the oversaturation of game consoles and bad games on the market threatened to end the industry as a whole. Now however, it cannot be denied that the video game industry is a multibillion-dollar juggernaut in the entertainment sector. The greatest proof of the current popularity of videogames would have to be videogame developer Rockstar's Grand Theft Auto V. Grand Theft Auto V alone made six billion dollars as of 2018, making it the best-selling *anything* in entertainment (Cherney, 2018). Videogames have become popular enough that ESPN even covers esports (high-level, competitive videogames) now. Since so many people love playing videogames it might make sense that there also exist people who enjoy watching other people play video games; enough that it can be someone's career now. But how did we get here? What events shaped the creation of the YouTube "Let's Player" actor network?

Background on the Sociotechnological System

People can turn playing videogames into a career with the use of ad revenue and viewer-donations on YouTube. This is possible because of the actor network made up of the viewers of YouTube, the owners of YouTube, the content creators (specifically Let's Players in this case), and the advertisers. The system works as such: the owners made and host the platform, the creators upload videos to the platform, the advertisers pay to have their ads appear on the videos, the viewers watch the videos and, along with it, the ads, and the creators and owners share revenue made from the price per view of hosting the advertisements. Every so often however, an actor will take actions that cause significant reactions and it is these events that I aim to investigate. For example, whenever the owners

implement new policy on what videos are eligible to host ads, when the owners change the site layout, when the owners change algorithms that determine what videos are suggested to the viewers, or when advertisers leave the network in response to some action made by a creator.

Data Collection and Analysis

By far the most useful source of data for the major developments in YouTube's history, will be public statements given by Google themselves detailing policy changes, and then public statements given in response to those policy changes. I'll need to look at YouTube's documented algorithm changes to show how they've favored Let's Players. I'll be looking at how major game releases corresponded with subscriber influxes for those covering those games, and the effects, as long as I can find subscription statistics to use. Another important thing I will be investigating is what effects new ways for Let's Players to make money (such as the addition of a donate button, YouTube Red, the paid-subscription "member" feature, and donations during livestreams) had on the Let's Players themselves, hopefully there will be statistics I can find for that, otherwise I'll only have personal accounts from Let's Players and news websites reporting on it to rely on.

Conclusion

By the end of this I will hope to have laid out a timeline concerning the history of Let's Plays on YouTube. Not just a timeline though, but a web of actions and reactions made by the members of the actor network that allows for Let's Players to not only exist, but be popular enough to be able to make a living off of recording video footage of them playing videogames while providing commentary. I will also have a working app for making and comparing Warframe builds. I will hopefully have helped show that videogames are not a

waste of time. There is money in making them and, for some, even money in playing them. Also, not all software needs to be for some critical system to help people, a simple companion app for a videogame can be useful to millions of people.

Works Cited

Bailey, D. (2019, March 25). Warframe added nearly 12 million players in a year. Retrieved December 13, 2019, from PCGamesN website: https://www.pcgamesn.com/warframe/warframe-player-count

Cherney, M. (2018, April 9). This violent videogame has made more money than any movie ever—MarketWatch. Retrieved October 18, 2019, from MarketWatch website: https://www.marketwatch.com/story/this-violent-videogame-has-made-more-money-than-any-movie-ever-2018-04-06

Duggan, M. (2015, December 15). Gaming and Gamers (p. 23). Retrieved from Pew Research Center website: https://www.pewresearch.org/internet/2015/12/15/gaming-and-gamers/