

Illinois State University

ISU ReD: Research and eData

Theses and Dissertations

2019

Toward a Theory of Procedural Rhetorical Systems: Demonstrations of Player Agency in Uptake of Rules in Video Games

Sydney A. Klem

Illinois State University, saklem@ilstu.edu

Follow this and additional works at: <https://ir.library.illinoisstate.edu/etd>



Part of the [Communication Commons](#), and the [Rhetoric Commons](#)

Recommended Citation

Klem, Sydney A., "Toward a Theory of Procedural Rhetorical Systems: Demonstrations of Player Agency in Uptake of Rules in Video Games" (2019). *Theses and Dissertations*. 1167.

<https://ir.library.illinoisstate.edu/etd/1167>

This Thesis-Open Access is brought to you for free and open access by ISU ReD: Research and eData. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of ISU ReD: Research and eData. For more information, please contact ISUREd@ilstu.edu.

TOWARD A THEORY OF PROCEDURAL RHETORICAL SYSTEMS:
DEMONSTRATIONS OF PLAYER AGENCY IN
UPTAKE OF RULES IN VIDEO GAMES

SYDNEY A KLEM

121 Pages

This paper expands Ian Bogost's (2007) procedural rhetoric by broadening the rhetorical view of games to encompass the arguments that they make not just about the material, but about themselves. The theory of procedural rhetorical systems (PRSeS) views game systems as arguing toward how players should be following their rules, and like in any form of rhetoric, players possess agency in how they take up these arguments and how closely they follow rules. To demonstrate this, this paper analyzes a specific game, the 1996 platformer *Super Mario 64*, alongside various digital artifacts demonstrating how players have taken it up, including videos, forum discussions, wiki entries, and comments. This paper divides the different ways players can view PRSeS into three uptake lenses (ULs), which are standard, speedrunner, and modder uptake. Where standard uptake represents taking up a game's PRSeS according to their exact argument, speedrunner and modder uptake represent taking them up in alternative ways, either with intent of beating the game as fast as possible or with the knowledge that rules can be modified and even transplanted from one place to another. These varied ULs prove that game rules are argued to players via PRSeS and that players have agency in how they take them up.

KEYWORDS: Digital rhetoric, procedural rhetoric, video game, speedrun, uptake

TOWARD A THEORY OF PROCEDURAL RHETORICAL SYSTEMS:
DEMONSTRATIONS OF PLAYER AGENCY IN
UPTAKE OF RULES IN VIDEO GAMES

SYDNEY A KLEM

A Thesis Submitted in Partial
Fulfillment of the Requirements
for the Degree of

MASTER OF SCIENCE

Department of English

ILLINOIS STATE UNIVERSITY

2019

© 2019 Sydney A Klem

TOWARD A THEORY OF PROCEDURAL RHETORICAL SYSTEMS:
DEMONSTRATIONS OF PLAYER AGENCY IN
UPTAKE OF RULES IN VIDEO GAMES

SYDNEY A KLEM

COMMITTEE MEMBERS:

Erika Sparby, Chair

Joyce Walker

ACKNOWLEDGMENTS

I thank Erika Sparby for being an excellent and supportive chair and providing me with fruitful research directions and great feedback, Joyce Walker for support and peer editing, Justin Charron for advice and beta reading, and Brittany Larsen for sources and research directions.

S. A. K.

CONTENTS

| | Page |
|---|------|
| ACKNOWLEDGMENTS | i |
| FIGURES | iv |
| CHAPTER I: INTRODUCTION | 1 |
| Overall Thesis Purpose | 1 |
| Review of Literature | 4 |
| Procedural Rhetorical Systems (PRSeS) | 6 |
| Uptake Lenses (ULs) | 8 |
| Game Genres, PRSeS, and ULs | 11 |
| Super Mario 64 | 12 |
| Methods and Methodology | 13 |
| Summaries of Chapters | 15 |
| CH 2: Standard Uptake | 15 |
| CH 3: Speedrun Uptake | 16 |
| CH 4: Modder Uptake | 17 |
| CH 5: Future Implications | 17 |
| Benefits of Study | 18 |
| CHAPTER II: STANDARD UPTAKE | 21 |
| Chapter Introduction - Standard Uptake | 21 |
| Rules and Progression in <i>SM64</i> | 23 |
| <i>SM64</i> 's PRSeS Arguing Toward Rules | 31 |
| Standard Uptake of <i>SM64</i> 's PRSeS | 33 |

| | |
|---|-----|
| Cheating Versus Standard Uptake | 36 |
| Player Motivations for Adhering to Standard Uptake | 40 |
| CHAPTER III: SPEEDRUN UPTAKE | 46 |
| Chapter Introduction - Speedrun Uptake | 46 |
| The Checkpoints and Ultimate Win-State of <i>SM64</i> | 48 |
| Speedrun Categories and Rulesets | 52 |
| The BLJ - <i>SM64</i> 's Biggest Skip-Enabling Glitch | 56 |
| Speedrun Uptake and Full-Area Skips | 59 |
| Speedrun Uptake Motivation | 65 |
| CHAPTER IV: MODDER UPTAKE | 69 |
| Chapter Introduction - Modder Uptake | 69 |
| ROM Hacks - Mod Type Pertaining to <i>SM64</i> | 73 |
| Rules, Progression, and PRSes in <i>Super Mario Star Road</i> | 78 |
| Standard and Speedrunner Uptake of <i>SMSR</i> PRSes Based on <i>SM64</i> | 83 |
| Modder Uptake | 86 |
| Modder Uptake Motivation and Motivation of Mod Players | 89 |
| CHAPTER V: FUTURE IMPLICATIONS | 91 |
| Chapter Introduction - Conclusion and Future Implications | 91 |
| Implications of Standard Uptake - Rules and PRSes for a Series | 93 |
| Implications of Modder Uptake - PRSes and Assets in Mashup Hacks | 100 |
| Self-Imposed Challenge - Uptake Lens for Future Research | 107 |
| REFERENCES | 112 |

FIGURES

| Figure | Page |
|--|------|
| 1. <i>SMB</i> progression on-screen | 24 |
| 2. <i>SM64</i> progression diagram (visual) | 26 |
| 3. <i>SM64</i> progression diagram (text) | 28 |
| 4. <i>SM64</i> progression on-screen | 29 |
| 5. <i>SM64</i> progression as explained in the instruction manual | 30 |
| 6. Content creator Flapper realizes what <i>SM64</i> is arguing he should do | 35 |
| 7. Big star door of area 1 | 50 |
| 8. Keyhole door leading to area 2 | 50 |
| 9. <i>SM64</i> 's page on speedrun.com with the 120 Star category tab open | 53 |
| 10. <i>SM64</i> 's hub area viewed as a wireframe | 57 |
| 11. Xiah using a BLJ to perform the area 1 skip | 62 |
| 12. A comment on a <i>SM64</i> speedrun showing speedrun uptake by non-speedrunner | 68 |
| 13. <i>Half-Life 2</i> vs <i>MINERVA</i> | 70 |
| 14. Toad's Tool 64, a modification tool for <i>SM64</i> | 75 |
| 15. <i>Kaizo Mario 64</i> is a ROM hack that increases <i>SM64</i> 's difficulty | 76 |
| 16. A screenshot of a level from <i>Super Mario Star Road</i> | 77 |
| 17. Compared to a screenshot of Bob-Omb Battlefield from <i>SM64</i> | 77 |
| 18. Compared to a screenshot of Princess Peach's Castle from <i>SM64</i> | 77 |
| 19. <i>SMSR</i> progression diagram (text) - compare to Figure 3 from CH 2 | 80 |
| 20. A big star door in <i>SMSR</i> (left) vs <i>SM64</i> (right) | 87 |
| 21. Mario jumping at a question block to get a power-up | 95 |

| | | |
|-----|---|-----|
| 22. | Filling unfilled block outlines in <i>SMW</i> | 97 |
| 23. | Like in <i>SMW</i> , <i>SM64</i> 's colored blocks start out unfilled | 98 |
| 24. | A lonely colored block found in <i>SM64</i> | 99 |
| 25. | <i>SMB64</i> vs. <i>SMB</i> | 101 |
| 26. | Pan's PU map and multi-display view from his video | 110 |

CHAPTER I: INTRODUCTION

Overall Thesis Purpose

In 2007, academic and video game designer Ian Bogost published *Persuasive Games*. Among the lasting and impactful concepts the book introduced was that of “procedural rhetoric,” a form of persuasion that works through interactive processes, including those in video games. At the core of Bogost’s procedural rhetoric was games’ ability to convey a systemic argument. Video games make these sorts of arguments by simulating systems that exist in the real world, then encouraging people who play them to interact with the simulated systems in a specific way. Based on how a game is designed, interactions with simulated systems can function persuasively. As players interact with them, they are left with an impression of a system’s values or purpose.

I absolutely agree with Bogost that in the design of video games there exist ideological, argumentative mechanisms that can efficiently convey ideas to audiences playing those games. Regardless of whether argument toward or reinforcement of ideology in a game is intentional, there’s no question it exists. As Bogost states, games like *Grand Theft Auto: San Andreas* (2005) can convey arguments based on how they represent characters who belong to minority groups. The player controls an African-American protagonist hailing from a low-income environment, and regardless of how much money they accrue, fast food remains the only consumable item. And on the topic of fast food, games like *McDonalds Videogame* (2006) argue about the poor, unsustainable conditions of that industry by structuring their systems in a way that players are incentivized to make morally questionable decisions to keep their virtual business functioning. Through the design of simulated systems, an argument is reinforced by way of player interaction, since players are encouraged to interact with those systems in a certain way.

From my experience with games and my connection to the communities around them, I've realized that there are dimensions of procedural rhetorical theory which could be expanded. This is not to say that Bogost's theory as it currently stands needs to be completely discarded. However, with the increasing complexity of video game design and its associated subcultures, there is room for expansion of procedural rhetoric as a model for examining rhetoric in games, not just in terms of how they can argue about the material world, but about themselves as well.

Bogost (2007) shows the difference between games arguing about reality and themselves. In his fifth chapter, he makes the claim a successful advertising video game, or "advergame," would be one that uses interactivity to argue to a consumer they want or need a certain product. Being able to try out the product for oneself in a game which simulates using it, for example, would argue that product's usefulness. A game where a player uses a vacuum cleaner to clean a room as quickly as possible could gamify the act of using the product, and in addition, argue toward its advantages for cleaning by creating gameplay scenarios highlighting those advantages. It would be like the video game equivalent of an infomercial demonstrating a product's features. Unsuccessful advergames, on the other hand, will fail to make a link to consumer wants or needs, thus failing in a connection to reality. These advergames usually only contain references to their brand in the form of visual elements pasted onto an otherwise unrelated gameplay experience; these advergames are, ironically, also the majority. Bogost sums up his thoughts on the matter:

"The contemporary approach to advertising games relies on the game experience as an end in itself rather than as a bridge to activities in the material world, making these advertisements simulations in Baudrillard's sense of the word - copies with no original, fantasies for a world that doesn't exist." (pg. 163-164)

As a self-professed lover of video games himself, Bogost obviously doesn't mean to attack the majority of video games for being simple trifles caught up in their own little worlds. Many of his points involve commercially successful games, such as the aforementioned *GTA*, alongside the seemingly-innocent Nintendo life simulator franchise *Animal Crossing* (pg. 267). Bogost simply focuses on how the rhetoric in these game worlds can influence the material one. Yet this still leaves an open question - what about those "fantasies for worlds that don't exist," the "copies with no original?" If procedural rhetoric can be assumed to exist in those worlds too, what does it argue to players about the ways that their rules function?

This is where I hope to enter this discussion in order to start addressing those questions. My view is that procedural rhetoric in games can make arguments that don't just influence how someone sees the world *outside* of games, but how they see the world *inside* of games as well. Expanded procedural rhetorical theory could open interesting discussions on how video games can make these sorts of arguments, along with the varied ways players might take them up. Ultimately, the rules of video games are not set in stone, but open to interpretation by players, whose ability to experience varied uptake of those rules grants them agency.

At its core, my argument has two parts: rules in video games are enforced rhetorically, and because they are enforced via procedural argument, players choose how they take them up. For this reason, in addition to arguing that rules in video games are indeed rhetorically enforced, I am also calling for, to use some gaming terminology, an "update" of procedural rhetoric theory. I am arguing toward a *theory of procedural rhetorical systems* (PRSeS), which future scholars, gamers, and any one else interested, can utilize to identify how games argue toward their rules, alongside the many different ways players can take them up.

Review of Literature

This project primarily builds on the arguments originally put forth in *Persuasive Games*. Early in the book, Bogost (2007) explains procedural rhetoric by using the history of rhetoric, drawing parallels between the designs of simulated systems in games and rhetorical techniques. Within a game, the subjective virtual space wherein the player is allowed to make their decisions, with feedback given by ideological framing of that space, is the procedural rhetoric (p. 29-31). Bogost likens it to the Aristotelian enthymeme, where a rhetorical proposition is left unfinished, requiring their audience to “fill in the blanks,” feeling like they came to a certain conclusion. However, their experience of autonomous problem-solving is illusory and is designed as such; the conclusion that they thought they reached was one that a rhetorician led them to (p. 43). Conversation on procedural rhetoric is consistent in most of Bogost’s work (2005, 2008, 2010), as is emphasis on how persuasive games argue ideological notions relating to real world issues.

Some sources make use of procedural rhetorical theory in the way that Bogost intended, and in doing so, show how his idea can be taken up and utilized in a number of different ways. Yet they also hint that procedural rhetoric could tackle topics its current form is not designed for. Klabbers (2011) discusses how persuasive games could fail in their aims at persuading players due to the varied uptakes a player may have, such as focusing on mechanics rather than message. Sicart (2008) argues that, thanks to the rhetoricity of news media in delivering information, “newsgames” are always persuasive by design, but rely on game mechanics for that persuasion. Harper (2011) talks about how games can use procedural rhetoric teach cultural value systems, but maintains that the mechanics conveying that rhetoric are autonomous in their own right.

Procedural rhetoric also has many critics, and most of them appear frustrated with what they perceive to be limitations of procedural rhetoric. In *The Judgment of Procedural Rhetoric*,

Simon Ferrari (2010) argues that the “game space is not merely the place where (ethical) dialectic occurs; it... embodies a ruleset (as) it... directs the flow of play.” In similar fashion, Miguel Sicart in *Against Procedurality* (2011) argues that procedural rhetorical theory, in its current form, has limits in both ideological and game design analysis despite having potential, due to a lack of discussion on how game mechanics can be taken up separately from ideology. Countless other publications have critiqued procedural rhetoric as well: Paul’s *Optimizing Play: How Theorycraft Changes Gameplay and Design* (2011), Matheson’s *Procedural Rhetoric Beyond Persuasion* (2014), and Ferrara’s *Argumentation, Procedurality, and the Lie of Gamification* (2013) to name a few. In entering the criticism conversation on procedural rhetoric, I work to expand procedural rhetoric to address critics’ concerns about the theory’s limitations. PRSes can encompass arguments games make about both the real world and their own rules, demonstrating Ferrari’s “dialectic” by showing the how players take up in-game rhetoric.

Of course, even with the highly varied uptakes procedural rhetorical systems can have, games are with designed rules in mind, and much work has been done on how they are taken up. In *What Video Games Have to Teach Us About Learning and Literacy*, Gee (2003) draws connections between players learning rules of games and students learning in classroom settings. He argues both cases see maximum success in student/player retention when they’re challenged, made to exert effort in digesting what they take up and arranging it in terms they understand. This aspect of game literacy has been cited by numerous sources and remains a popular point of discussion in game studies (Chen, Dutchuk, 2009; Lave, 2010; Hidayah, 2012).

I would agree with Gee (2003) and others who’ve cited him that this occurs with players. Yet like with any type of learning, there’s a chance game rules will be taken up in alternate ways, and this does not necessarily mean that the players technically “failed” to learn the game’s rules.

Some players simply take up game rules in a way where despite understanding their parameters, they decide to play outside them for enjoyment, for challenging themselves, or for other reasons. This is where a rhetoric-based lens could have its own unique benefits for studying game rules, as viewing game rules as a being delivered through PRS-based arguments garnering multiple possible responses would shed light on the complexity of player uptake of video games.

Procedural Rhetorical Systems (PRSes)

A PRS is any element of a video game's design which, through direct player interface, can convey an argument about how players are supposed to be playing it according to its rules. PRSes could be anything from an in-game tutorial that directly explains "how to play" the game to a progression barrier that argues to players that they must take specific action to get around it. Bottomless pits in *Super Mario Bros.* (1986) provide a perfect example of the latter type of PRS. A bottomless pit serves as a *rhetorical system* which suggests that there is some way to cross it. The *procedural* part comes in when players experiment with controls and learn to jump over pits. Thanks to that bit of player interaction, the procedural rhetoric succeeds in its intended purpose, reinforcing in the player's mind one of the game's rules - when encountering a pit, jump over it. Impressive about this pit, and other PRSes like it, is that it does not involve verbal explanation. While PRSes certainly can involve language directly explaining to players how to play a game, the mere existence of the bottomless pit as an object players can interact with is often enough to persuade them on how the game's rules should function, or how the game is meant to be played.

It is because of objects like the pit that rules are something anyone who's played a game, and accrued at least some experience in it, can usually explain in surprising amounts of detail. Asking someone how *Super Mario Bros.* is supposed to be played, for example, will most likely yield answers relating to jumping over hazards and progressing through levels. Asking several

other players a similar question about the game will likely yield similar answers. The way that the game's rules are described by those players will seemingly be influenced by something they likely never noticed - the invisible guiding hand of PRSes.

This is important because unlike sports or board games, which require an external agent (such as another person or manual) to impart rules, video games argue about rules autonomously. A video game can serve as not just the board or playing field, but the manual or teacher as well, making an argument about how its pieces should be played with. The act of playing the game, and discovering how the game wants to be played, are closely intertwined because of this fact. This could also be said of traditional games, which can hint at their rules through, for instance, the design of their physical game boards. However, video games can do this with greater ease, providing direct feedback to player interaction and rewarding them for overcoming challenges. It's possible, as Gee (2003) demonstrated, to equate this sort of feedback to a form of learning, since games are almost always designed to reward players for following the rules as intended. However, my emphasis remains on mechanical systems as rhetorical rather than as educational. The reason for this is simple: if there are commonplaces among players which reinforce how video games are meant to be played, then this notion cannot exist independent of its opposite. That is, of course, the way that games are *not* meant to be played.

My evidence for this lies in what is the bulk of my project - the ways PRSes are taken up. Since game rules are delivered to players via PRSes, this means that while some players might follow the PRS-argued rules without a second thought, others might take them up differently. They could even avoid following game rules in the exact ways the PRSes suggest they should while still managing to finish the game according to the rules those same PRSes argue toward. The fact that this is a possibility cements my argument that video game rules are delivered in a manner similar

to traditional games, where the rules are followed mainly because of convention. Therefore, the incentive for following video game rules is not due to hard, unbreakable coding, but rhetorical engagement between player and game.

Uptake Lenses (ULs)

The rhetorical relationship between player and game leads to many uptake lenses (ULs). ULs are essentially lenses players choose to wear when determining how they'll take up PRSes. However, to get some of those lenses, players might need to craft them using special knowledge, which cannot be found in most games by default. Such knowledge includes performing glitches, manipulating game code, and other acts which have potential to augment one's uptake of PRSes. Thus, while any player *could* possess any UL, some ULs may be more exclusive than others - and yet, for this exact same reason, the possibility for variation among ULs is endless. Therefore, to make the scope of this project manageable, I plan on defining and exploring three main ULs: standard, speedrun, and modder uptakes. Explaining these ULs is easiest using two premises: what performing the UL entails, and what the relation of the UL to PRSes is.

Performing standard uptake involves playing a video game as it is meant to be played. One playing a game in the intended way would avoid using glitches or cheating methods, attempting to stick to the rules that are clearly intended and conveyed within the game's design. Standard uptake may occasionally involve acts considered outside the rules, such as using cheats, walkthroughs, or other means. Yet these sorts of events often revolve around standard uptake, serving more as subcategories of it rather than breaking away from it to create their own ULs. Even if players following standard uptake might occasionally subvert PRSes, those players will, in the majority of cases, demonstrate behavior which steers them back towards closer adherence. Standard is the uptake lens that most, if not all, people who play a game will experience, arguably

standing as a prerequisite for understanding a game in its intended form. Therefore, standard uptake is defined by taking up PRSes in ways which resist subverting them.

Explaining the performance of speedrun uptake is best-done by describing speedrunners. As their name implies, speedrunners attempt to play through video games as quickly as possible. They are known for pushing games to their absolute limits in order to achieve world records, which often involves making use of glitches that allow games to be played faster than intended. Some speedrunners might modify game code in order to make games more speedrun-friendly, but usually, they stick to the standard game text, only using glitches present in its original code. Glitches that allow games to be played faster than intended usually fly in the face of their rules, and speedrun uptake thus often involves playing games in ways they are not meant to be played. Speedrun uptake is an experience in which the rules of video games needn't be strictly followed, even if they must be in some capacity in order for a speedrun to still have a definable goal.

This means that within PRS relations, speedrun uptake plays things more fast and loose. One could also argue that speedrun uptake interprets a game's arguments in a free-form manner, accepting the core argument which reflects the objectives of the game that determine completion, while simultaneously throwing out parts of that argument which restrict their rate of completion. Intriguingly, speedrun uptake also seems to have potential to find new, totally unintended PRSes. The ways speedrunners make use of glitches could be considered PRSes in speedrunning circles, because they define a set of rules that speedrunners follow which differ from the standard game. They turn rule-bending glitches into techniques recognized by speedrunners to expand the game, creating an additional set of PRSes that are usually invisible to those abiding by standard uptake. Speedrun uptake is thus defined by taking up PRSes with willingness to potentially subvert them,

and a willingness to see new ones, while simultaneously adhering to arguments posed by PRSes which determine how completion of a game is achieved.

Similar to speedrun uptake, modder uptake is easily explained by talking about modding. Modding is short for “modifying” and involves digging into a game’s code to make changes. Changes can range from cosmetic alterations to taking assets, such as pieces of existing levels, and rearranging them to create new levels which contain totally original layouts and challenges. Some modders create sequences of custom levels and package them as their own self-contained experiences, which can be downloaded and played by owners of the game they are based on. These are called “mods” and can sometimes contain just as much content as a full-sized game. However, mods are closely tied to the game from which their code originates, and many of them can be described as unofficial “extra chapters” of their game of basis.

For the above reasons, mods often carry over the same gameplay as their game of basis. One well-known mod is *MINERVA* (2005), which is based on the code of *Half-Life 2* (2004). *Half-Life 2* is a first-person shooter which emphasizes combat, exploration, and solving puzzles. *MINERVA* utilizes the same graphical assets, enemies, and general gameplay seen in *Half-Life 2*, but remixes them into a new series of levels. Watching *Half-Life 2* and *MINERVA* side-by side, one would see plenty of shooting, exploration, and the occasional puzzle, making the two of them look nigh-identical. The main difference between *MINERVA* and *Half-Life 2* are the levels. Yet those levels do such a good job of recreating *Half-Life 2*’s mechanics and general “feel” that, were they to be slotted into *Half-Life 2*’s existing suite of levels, they wouldn’t stand out at all. *MINERVA* breaks *Half-Life 2* into its component pieces and rebuilds them into a new experience, and because it owes so much to its game of basis, it copies its rules as well. Weapons, enemies,

objects, and other entities all function exactly the same in *MINERVA* as they do in *Half-Life 2*. Someone playing the mod would encounter the same PRSes as they would in the actual game.

It should be noted that the complexity of modder uptake in relation to PRSes is massive. Therefore, I exclusively examine modder uptake as it pertains to examples such as *MINERVA*, where mods are distinct entities that provide new experiences based entirely on an existing game. Mods showcase modder uptake of PRSes, which is an understanding similar to standard uptake, since one must know the PRSes of a game in order to know how to recreate them within a mod. Yet modder uptake doesn't end there; as with *MINERVA*, modders bring their own perspectives, sometimes adding to the code of the modded game with new mechanics which need new PRSes. Thus, modder uptake is defined by taking up the PRSes as a meta-argument, capable of being transplanted to other areas, while also being willing to add PRSes from any source available.

Game Genres, PRSes, and ULs

Of course, it must be acknowledged that video games are varied in genre and structure, and PRSes in different games and genres make completely different arguments from one another. Some games are more linear in design, while others attempt to give players a bevy of options. The aforementioned *Grand Theft Auto: San Andreas*, for example, is considered “open world,” situating it within a genre which emphasizes tackling missions in an open-ended environment. PRSes relating to mission objectives in *GTA* would argue to the player that they have options, such as the direction they approach the mission area from or the weapons and vehicles they use. In contrast, the aforementioned *Super Mario Bros.* is said to belong to the “platform” genre, which generally emphasizes straightforward progression across a series of levels in linear order. PRSes in these games challenge players to clear objectives with a more limited set of choices, such as how *SMB* challenges players to jump over obstacles without the option to “go around.” These two

genres are completely different in the type of experience they wish to offer players, and because of this, their PRSes can take different forms and make totally different arguments. Altogether, this means that the differences in game genres make studying PRSes complicated.

All this also means that when talking about ULs, focusing on the possible uptakes of one game will never yield the exact same results as talking about uptake in relation to another game. As meanings of PRSes change, so do meanings of ULs - especially in games where speedrunning and modding are built-in, such as *A Hat in Time* (2017). Considering variance of PRSes and ULs, a study of video games, PRSes, and uptake of them is best served through a more narrow focus. Focusing on one game and how its PRSes have been taken up allows for a more specific and nuanced understanding of PRSes in general.

Super Mario 64

Super Mario 64 (1996), the primary video game text that is analyzed within this project, is useful to the study of PRSes and how they are taken up by players for a number of reasons. For one, *SM64* belongs to a specific genre - the 3D “collect-a-thon” platformer (Wirtanen, 2017). Games in this genre usually present players with a central hub area they must progress through. However, in most collect-a-thons, progress cannot be made by just exploring the hub area itself. Hubs typically contain entrances to levels in which collectible objects can be found, and after collecting certain sums of these objects, players are able to move forward in the hub. This creates a clear system of progression via exploration and reward within the hub area, making the conventions of this genre fairly obvious since they are easy to spot in its very design. These conventions provide a clear structure for players to follow along throughout the game, while also arguing to players that they have agency in how they’ll handle objectives. Therefore, *SM64*’s

PRSeS are useful not only because they are very easy to see in the game's actual design, but because they are linked to my three ULs (standard, speedrun, modder).

SM64-related ULs have had plenty of time to develop examples proving their existence, partly thanks to the game's high popularity and status as a multi-million seller (O'Malley: 2015) and partly thanks to its age. *SM64* was first released in 1996 as a launch title for the Nintendo 64. Since then, the game's publisher, Nintendo, has re-released the game for many newer consoles. Beyond that, *SM64* is extremely popular in the emulation scene, allowing the game to be played (in a tenuously legal manner) on various other devices as well, including PCs, phones, and more. The number of times *SM64* has been played has resulted in numerous ULs demonstrated throughout the years, making the game ideal for structuring chapters about ULs around.

Methods and Methodology

Since my project centers on three ULs, I will be making use of a case study methodology. A case study (Heale, 2018) is one that usually focuses on "a person, a group of people, or a unit." This makes one ideal for analyzing ULs, as evidence for ULs is produced by people and groups. Case studies also have a tendency to discover connections between people and groups as well, even when those connections may not appear to be created by typical lines of communication. Regarding ULs, connections between different players can be drawn in how they take up PRSeS, especially if they show similar behavior in a particular game in the same situation in that game. This would suggest that ULs are definable with set boundaries in terms of what players take up, since multiple people can display similar behavior responding to the same arguments of PRSeS. And of course, this project focuses on people and groups demonstrating responses to *SM64* with particular attention paid to artefacts they've left behind.

Artefacts made by *SM64* players are the focus of this project's data collection methods. These are objects which provide evidence for the existence of ULs and are mainly found online. Each UL will be defined through evidence provided by artefacts of varying type and description. For standard uptake, artefacts include walkthroughs (guides explaining how to complete games), with attention exclusively (and naturally) centered on walkthroughs that were made for *SM64*. They can appear in print, video, and other formats. Additional artefacts include the game itself, along with gameplay videos that show how players make decisions in-game based on uptake. Finally, artefacts relating to cheating in games address how cheating is an offshoot of this UT. Those artefacts include more gameplay videos along with articles about cheat devices from digital and print publications.

Most speedrun uptake artefacts show evidence of *SM64* speedrunners subverting PRSes, including numerous speedrunning videos. Others show that speedrunners are aware of PRSes, and the rules that they enforce, even if speedrunners often choose to ignore numerous PRSes. These artifacts consist of descriptions of *SM64* speedrunning rule sets on various websites. Finally, additional artefacts show how this UL might be able to detect new PRSes within *SM64*. These include tutorials in video and text form which show how to perform glitches in the game, including specific spots where they are possible.

Modder uptake artefacts mostly show uptake of PRSes as transformable and transferable. Such artefacts include sites on which mods are hosted, which often include descriptions which verify developer intent in either recreating or subverting PRSes in *SM64*. Other artefacts demonstrate how PRSes from both *SM64* and from speedruns are recreated in *SM64* mods, further reinforcing the meta-textual understanding of PRSes which modder uptake represents. Such artefacts include walkthroughs on how to complete *SM64* mods and gameplay videos.

Summaries of Chapters

The following summaries detail what can be found in each of my project's chapters. There are three main chapters followed by a conclusion. Each of the main chapters is based around a UL and is structured to define that UL with the aforementioned artefacts as evidence. Chapter 2 will focus on outlining standard uptake, chapter 3 will focus on speedrunner uptake, chapter 4 will focus on modder uptake, and chapter 5 will conclude how ULs and PRSes work. The ultimate objective of these chapters is to use the varied ULs that determine player uptake to outline the functionality of PRSes, ultimately showing how they work to argue about game rules, and how players choose how they take up those rules in a form of rhetorical engagement.

CH 2: Standard Uptake

Anyone who has played *SM64* likely knows how to take up its PRSes the standard way. Speedrunners, modders, and other players all understand how the game is meant to be played, even if they don't engage with the game that way. Chapter 2 focuses on establishing clear parameters for standard uptake of *SM64*. The first step toward accomplishing this is figuring out how *SM64* is meant to be played. Attention is paid to how *SM64*'s rules and progression systems work, using evidence from the game itself alongside walkthroughs and gameplay videos. With rules and progression established, attention then shifts to how *SM64*'s PRSes argue toward following them. PRSes manifest in the game as everything from doors to collectibles, sending an obvious message for what players are meant to be doing. Videos are used, as they show players responding to certain events and objects in the game which lead them to play abiding by rules. Direct evidence of this kind shows how players take up PRSes when they are thinking in terms of standard uptake.

Chapter 2 also pays significant attention to how games are not meant to be played and what this might imply about the choice to agree to the standard uptake argument when playing. This end is satisfied by discussion of the use of cheat devices which can subvert *SM64*'s rules. With the right resources, it seems any player can take up the rules of any game in deviant ways, furthermore suggesting that rules in video games are not followed by necessity, but by choice. Convention established via a sort of rhetorical contract appears to be the reason that rules are abided by in most instances, as without acting within limitations, the game wouldn't be a game. Since acting in limitations is a choice, enforcement of rules in games is rhetorical, persuading players to play a certain way via PRSes.

CH 3: Speedrun Uptake

Establishing clear parameters for speedrun uptake is the focus of my third chapter. Speedrunning is established first, with special focus on the goal of speedrunning as a whole: completing games quickly. This transitions into how speedrunners determine their guidelines partly through use of the game's PRSes, and how they solidify this in speedrunning categories. Speedrunning categories suggest that those with this UL must acknowledge standard uptake. Regardless of whether some PRSes are ignored or adhered to, the ones which determine the game as being "complete" are followed; without them, speedrunning would be meaningless.

Another important discussion related to speedrun uptake revolves around glitches. *SM64* speedrunners make use of a wide variety of glitches to skip massive chunks of the game, such as the "backwards long jump" (BLJ), which permits one to phase through doors and walls. Glitches like the BLJ have been tutorialized, such as in videos and text on speedrunning sites, which show players how to perform them. Chapter 3 reveals that those who use speedrun uptake can follow their own rulesets, even coming to perceive totally new, unintended PRSes.

CH 4: Modder Uptake

Chapter 4 invokes discussion of both speedrun and standard uptake as it discusses the complexities of modder uptake. Modder uptake is the toughest UL to establish parameters for, which is why I've chosen to focus on one aspect of it connected to speedrun and standard uptake. This element of modder uptake is the ability to take up PRSes in a way where it is understood that those PRSes are transferable and remixable in mods. Chapter 4 builds toward an understanding of this by discussing mods which demonstrate a clear intention in recreating *SM64*'s PRSes in new contexts.

Chapter 4 also connects to chapters 1 and 2 in how it discusses the overlap between modder uptake, standard uptake, and speedrun uptake. Modders must comprehend standard uptake in order to know that *SM64*'s base PRSes can be lifted and re-used in different contexts. Mods must anticipate how a player with standard uptake encounters and takes up PRSes in order to remake them in a way that they take them up the same way they would in the actual game. This suggests those with modder uptake are technically capable of not only "reading" *SM64*'s PRSes to gain a greater understanding of them, but creatively "writing" with them as well.

CH 5: Future Implications

If there is one thing I would like to emphasize about PRSes in my closing chapter, it would be their overall versatility. PRSes are not set to a certain kind of uptake. The ULs I have identified possess fixed, identifiable characteristics, but they merely represent certain styles of uptake that take place in the moment that someone plays a game and absorbs information about its systems. Any person can experience any uptake lens at any time, and few people experience just one at any given time. A player who has watched some speedruns of *SM64* might notice opportunities for glitch use even if they do not know how to perform them - a speedrunner who skips past vast

chunks of *SM64* understands that what they are doing is not how they are meant to be playing. Players have unlimited lenses to slide in and out when enjoying a game, and this makes their uptake both immediately understandable and endlessly complex.

Benefits of Study

All the elements in this project contribute to the objective of building on Bogost (2007), primarily in the area of examining how game mechanics affect player uptake of games as texts. This will prove beneficial for academic courses that mix game studies with rhetoric especially. University-level rhetoric courses have experimented with having students analyze game design, focusing particularly on how persuasive games can convey arguments in ways Bogost describes. In the case of some of these courses, such as one referenced in a case study by Colby (2014), something interesting happened with how students took up persuasive games.

Many of the students who played persuasive games experienced some unexpected uptake. With *The McDonald's Videogame* (2007), many students sympathized with the corporation, because the game's structure and high difficulty "portrays running a company as quite difficult" (Colby, pg 45). Considering the reason the game is difficult lies primarily within its game design, something seems to be happening with player uptake of PRSes in a primarily mechanical sense. Perhaps students' sympathy was generated, at least in part, by the game's high difficulty level, which forced them to learn and master the game's mechanics, leading to a satisfying experience. *The McDonald's Videogame's* mechanics, isolated from any commentary that they might make, might be seen as worthy of investment by players who wish to simply "beat it" as a video game. Those players might then have a positive experience overcoming the challenge of the mechanics, intaking PRSes as they learn the rules and figure out how to effectively run their virtual business.

Player attachment to *The McDonald's Videogame's* mechanics and PRSes, fleeting as it may be, could cross over into a feeling of sympathy for the corporation in real-world terms.

In this instance, it might have been useful to teach students about the difference between procedural rhetoric arguing on game rules and procedural rhetoric arguing on real-world ethics. The ways players take up mechanics and rules are largely dependent on their game experience - experience which dictates the ULs they apply like lenses while they are actively playing a game. Those with gaming experience are likely to take up and comprehend PRSes as mechanics first, understanding a game's rules through close observation of how its structure supports those rules. This has been demonstrated in numerous studies and can cause players to focus more on how to overcome video games as mechanical challenges rather than as narrative or ideological puzzles. Mackey's (2007) study of children's uptake of media based on experience shows this (p. 81-102). Jack, the child Mackey believed had the most video game experience among her sample group, played the narrative-driven *Starship Titanic* (1998) with Anita, who had less game experience. Anita focused on the game's story, while Jack focused on figuring out its mechanical structure. As Mackey herself states, "Jack was looking for rules; Anita was looking for content."

Uptake of PRSes influenced by mechanics clearly plays a significant role in how games are taken up in general, yet those uptakes could stand to be better-integrated into Bogost's theory. This is where PRSes are handy, as they can still account for procedural ideological arguments, but in addition, they recognize the influence that mechanical uptake has on perceptions of games. Having a theoretical methodology with this level of versatility would be extremely useful for teaching students about how games can convey arguments, which would likely lead to clearer, richer conversations about procedural rhetoric as a whole. As a bonus, tech-savvy students who

are experienced with games (whose numbers increase by the day) would be able to have greater input in rhetoric classrooms, contributing to more collaborative learning spaces.

This project works toward an expansion of Bogost's theory for this and other reasons. Bogost provides an excellent tool for examining video games from a rhetorical point of view, which has several benefits for game studies and rhetorical studies alike. One goal of PRSes, then, is to sharpen this tool, making it capable of taking more game design elements into account, including those which might not have an obvious rhetoric function behind them at first glance. PRSes show how the seemingly un-rhetorical rules and limitations that are present within games, which appear to exist only to guide players to playing a video game as it is meant to be played, are actually argumentative in nature, persuading players to follow rules by their own free choice.

CHAPTER II: STANDARD UPTAKE

Chapter Introduction - Standard Uptake

Standard uptake entails following a game's rules exactly according to arguments made by its procedural rhetorical systems (PRSeS). Among the three uptake lenses (ULs) in this project, standard uptake represents how the majority of players will take up a game's PRSeS and play it. For this reason, another way of thinking of standard uptake is taking up a game's PRS arguments according to how the game is meant to be played.

In this chapter, standard uptake is defined through a detailed analysis of *Super Mario 64* (or *SM64*) and what it means for this particular game's PRSeS to be taken up in the standard way. However, two factors need to be accounted for in regards to the methods behind this analysis. The first factor relates to evidence for standard uptake and variance in player-game interaction. Uptake evidence will mainly come from gameplay videos which show first-time *SM64* players, as they demonstrate how players figure out how to view PRSeS arguments using standard uptake. However, while standard uptake itself is a structurally predictable lens most players will learn, the way that they initially discover it when playing *SM64* for the first time can vary significantly. Some players might come in with knowledge of *SM64* and perceive standard uptake quickly, while those without such knowledge might take longer to see how the game wants them to play. This leads to variance in first-time player action, with some who play *SM64* wandering around before figuring out what they are expected to do, while others discern their objectives faster. Thus, this chapter focuses on a small number of specific videos which feature first-time players. Furthermore, this chapter narrows its focus onto certain sections within these videos which show the moment where players take up and accept the argument of a specific PRS.

This brings me to the second factor, which relates to the emphasis on a specific PRS. Like with the vast number of different paths players may take to finally perceive standard uptake, PRSes are incredibly varied and can appear in practically any audiovisual form within the game. Everything from environment design to sound effects to items can create procedural arguments, with the player having to agree with them both cognitively and through the action of playing. Therefore, to get the most out of the analysis within this chapter, only a single PRS is focused on, with the aforementioned videos each containing footage which demonstrates uptake of this PRS. With this chapter's specificity, a clear picture of standard uptake in *SM64* comes into view.

Even with this focus, however, it is necessary to situate standard uptake in a certain way. Since standard uptake entails following *SM64*'s PRSes and playing the game as it was intended, this means that the PRSes are arguing toward the game's core rules and its progression system. Thus, *SM64*'s PRSes, and standard uptake of them, are best-understood by a top-down approach. This entails starting at the "top," with the actual rules and progression which govern the game, then moving "down" to the aforementioned specific PRS and looking at how players take it up. This will make clear the functionality of PRSes in *SM64* as facilitators of progression and rules, while also making clear the kinds of rules and progression systems standard uptake highlights, along with the wider relationship between the players, standard uptake, and the rules.

After standard uptake is defined, this chapter discusses why players play according to it. With the amount of choice players seem to have in determining their uptake of *SM64*'s PRSes, why do most of them follow standard uptake instead of any of the other lenses that are available? This question will be answered through discussing cheating as a mutation of standard uptake, along with elements that make standard uptake appealing for player motivation.

Rules and Progression in *SM64*

Progression systems represent how players move through a game to reach an end goal. Though similar on the surface, progression systems are entirely distinct entities from PRSes. PRSes make arguments to players about how they should be playing and progressing in a game. Progression systems govern how the player's advancement through a game can be measured. Thus, a PRS could argue to a player they are meant to solve a maze while staying in its bounds. Progression would be each step the player takes which brings them closer to the maze's end.

Single-player games like *SM64* are easy to comprehend in terms of progression systems, since they often work off a series of telegraphed steps, or checkpoints, laid out in linear order. Telegraphing needs “checkpoints that a character must reach to advance” (Technopedia, 2019). Checkpoints in games are usually represented by visuals which show how far a player has gone.

Two checkpoint types are relevant - environmental and user interface (UI) checkpoints. Environmental checkpoints are, unsurprisingly, those appearing within a game's simulated space. In *Super Mario Bros. (SMB)* for example, a very important environmental checkpoint is the flag¹. In Figure 1, it is clear that the player has made progress because they are touching the flag, which helpfully indicates they have reached the end of the current level. Though certainly useful, environmental checkpoints have limits when on their own thanks to the nature of game visuals. *SMB*'s levels are all built off of the same sprite² template for bricks, bushes, and even the ground, making it hard to tell levels apart. Thus, while the player has clearly reached the end of *a* level, *which* level they've completed, and how far they are in the overall game, is far less obvious. Screenshots of levels at the game's beginning and end could look nearly identical.

¹ In *Super Mario Bros* (1985), the player progresses through two-dimensional levels moving left to right. Each level ends with a flag, which the player touches to end the current level and move to the next one.

² Pixelated building blocks of game graphics in older two-dimensional titles like *SMB*.



Figure 1. *SMB* progression on-screen.

For determining progress in the overall game, user interface (UI) checkpoints are useful. UI elements usually stay on the edges of the screen, displaying status information such as health, items collected, and so on. In older games, UIs also tend to display the level that a player is on. *SMB* does this in a very simple way. The game contains eight worlds split into four levels each. Players start on World 1-1 - or “world one of one,” after which they play 1-2, 1-3, and 1-4, followed by 2-1, 2-2, and so forth. This info is shown at the top of the screen in the level counter, and to measure how far someone has gotten in *SMB* overall, one need only look at said counter. Environmental checkpointing in Figure 1 shows a player having progressed in the current level, yet UI checkpointing reveals that in terms of overall progression, they are on the first level, 1-1. Progression is often best gauged by using a combination of environmental and UI checkpoints, since it can be difficult to determine how far a player is in a game with just one or the other.

Like *SMB*, *SM64* has environmental and UI checkpoints which communicate progression. However, *SM64*'s checkpoints are more abstract than those of its two-dimensional predecessor,

since how “far” a player is in the game can’t be determined by their distance to a singular flag, nor can it be determined by looking at the specific level they are on. This is best explained by comparing the two games directly to each other in terms of how their levels are sequenced.

The original *SMB* sends players on a linear path through each of its numbered worlds. There is obviously some variance in the order in which players might experience these levels, thanks primarily to the existence of warp zones³ which allow players to skip ahead in the game. Yet while levels can be skipped, they cannot be replayed or tackled outside a linear order. This, combined with UI and environmental checkpoints, makes progression effortless to perceive. There are so few variables that asking “how far” someone is in *SMB* can be determined by just glancing at the screen, which displays consistent environmental and UI checkpoints at all times.

The answer to “how far” someone is in *SMB*’s 3D descendant can be more ambiguous. *SM64*, like other 3D platformers (Cozic, 2012), does not start players on a 1-1 equivalent. Rather, players begin at Peach’s Castle, the game’s hub, an area which serves as a kind of lobby. The hub is not a level in and of itself, but rather a space where the levels can be accessed from. The hub is divided into four main areas. These areas are meant to be unlocked in a specific order, though the player is allowed move freely between them once they are available for exploration. Thus, environmental checkpointing is determined by the number of areas a player can access. This can be seen in Figure 2, where the hub areas are all unlocked in a clear and linear sequence, with each area also serving as a sub-hub of sorts for its own selection of levels.

³ Warp zones are hidden areas which contain passageways to various worlds. Players can use them to skip ahead in the linear sequence of levels, such as using the one in 1-2 to go to world 2-1, 3-1, or 4-1.

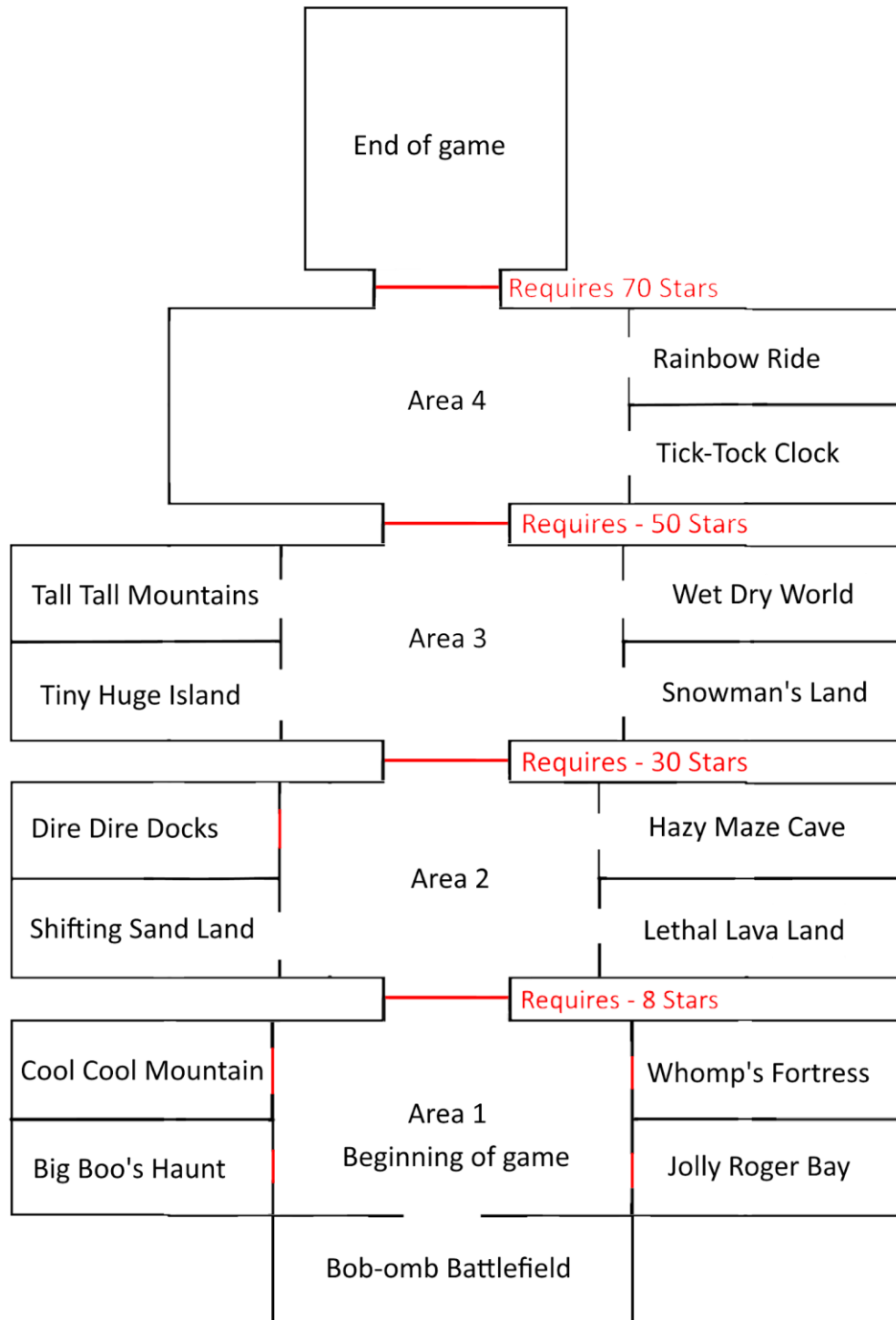


Figure 2. SM64 progression diagram (visual).

Unlocking all the areas throughout the castle means that a player can choose any level, and thus, unlike in the original *SMB*, levels can be both played over again and in various orders. This is the reason the clearest environmental checkpoint in the game is the player's current area - if a player is in the third of the four areas, it is safe to assume that they have made progress. However, if the player is in the first area, it's still possible to see that they have made progress.

The reason for this is because of *SM64*'s most important UI checkpoint - the star counter. Mentioned in Figure 2 and elaborated within Figure 3, *SM64*'s areas are divided by locked doors. In order to open those doors and make progress from area to area, the player must enter levels, which can be accessed via entrances that are found in each hub area. Within each of those levels, players will find power stars. With enough stars, new areas in the hub can eventually be opened, revealing new levels with more stars.

This is why figuring out progression in *SM64* is somewhat more complex than in *SMB*. Based on environmental checkpointing alone in Figure 4, the player could be at the start of the game with no progress as they could be ready to reach the end of the game whenever they want. However, their overall progress in the game can be easily determined thanks to their star count. With forty-one stars, they have access to areas 1, 2, and 3, marking significant progress.

SM64's environmental and UI progression create a visually communicated set of rules. Players are to progress through the castle's four main areas by visiting levels and collecting stars. This is obviously how *SM64* is meant to be played, further reinforced by its instruction manual, displayed in Figure 5; "The more Power Stars you collect, the more places you can go" (1996).

**Each level contains seven stars,
used to unlock more levels and areas**

| | |
|--|---|
| Area 1 Access requires - 0 Stars | Area 3 Access requires - 30 Stars |
| Bob-omb Battlefield Unlocked with - 0 Stars | Tall Tall Mountain Unlocked with - 0 Stars |
| Whomp's Fortress Unlocked with - 1 Star | Wet Dry World Unlocked with - 0 Stars |
| Cool Cool Mountain Unlocked with - 3 Stars | Tiny Huge Island Unlocked with - 0 Stars |
| Jolly Roger Bay Unlocked with - 3 Stars | Snowman's Land Unlocked with - 0 Stars |
| Big Boo's Haunt Unlocked with - 12 Stars | Area 4 Access requires - 50 Stars |
| Area 2 Access requires - 8 Stars | Tick-Tock Clock Unlocked with - 0 Stars |
| Lethal Lava Land Unlocked with - 0 Stars | Rainbow Ride Unlocked with - 0 Stars |
| Shifting Sand Land Unlocked with - 0 Stars | End of game Access requires - 70 Stars |
| Hazy Maze Cave Unlocked with - 0 Stars | |
| Dire Dire Docks Unlocked with - 30 Stars | |

Figure 3. SM64 progression diagram (text).



Figure 4. SM64 progression on-screen.

Considering *SM64*'s rules and progression system, it is easy to see how the game is meant to be played. Yet on the flipside, it's also very easy to see how the game *isn't* meant to be played. If a player accessed a later area but their star count wasn't high enough to have reached that area, it'd appear they hadn't played properly due to a mismatch of environmental and UI checkpoints. Anyone who saw a player in area four with a star count of zero would likely assume they found a glitch or used some kind of cheat device that enabled them to skip over needing to collect stars. Such thoughts would arise almost instinctively to those with significant knowledge of the game, yet in regards to procedural rhetoric, there remains a question of where this *meant to be played* notion originates from. What is the premise determining the conditions by which a player could be described as a cheater - or, in other words - playing the game as it is *not meant to be played*? The answer, of course, lies in a series of interactive arguments that the game conveys to players, but what is the structure of this argument, and what does it look like in action?

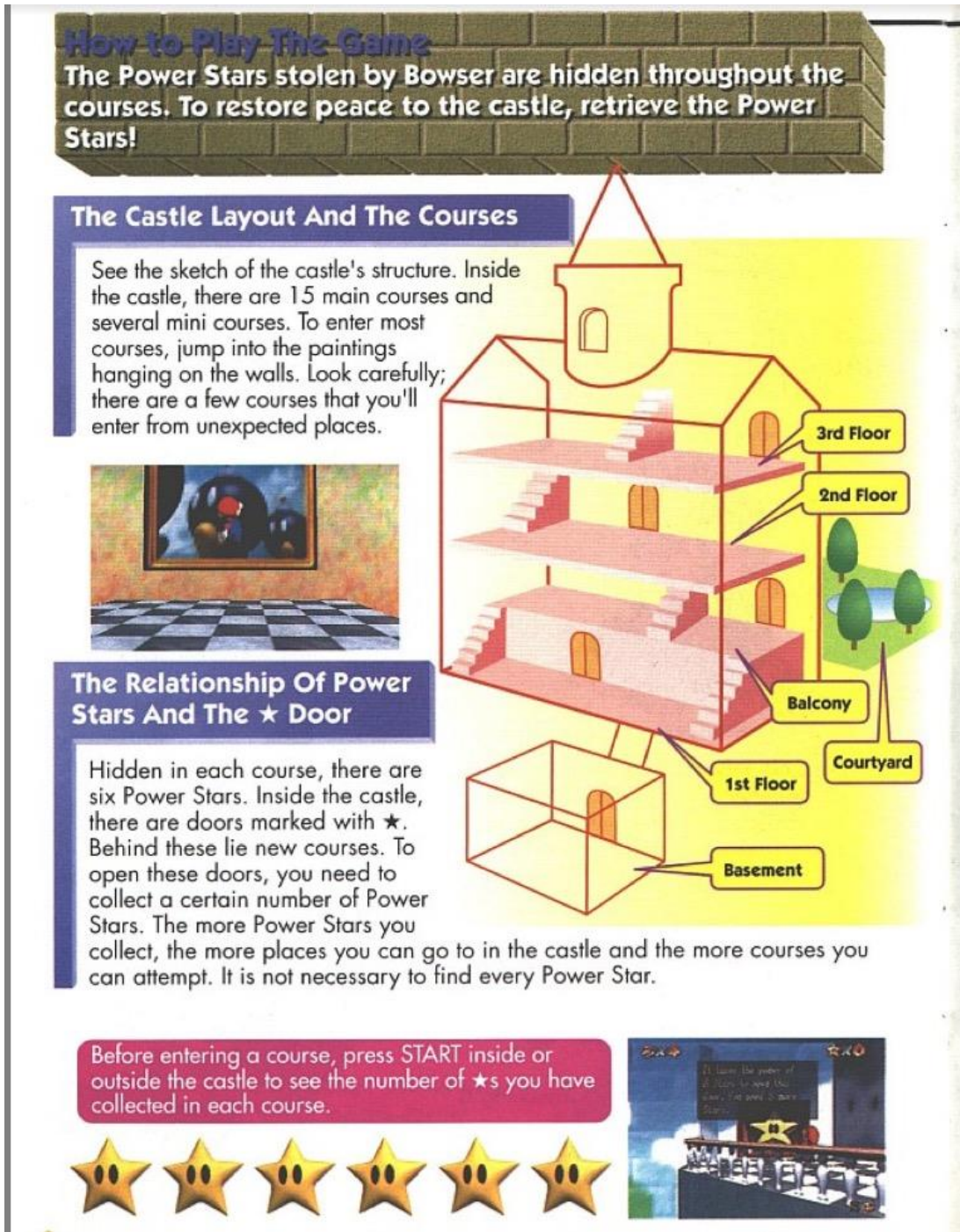


Figure 5. SM64 progression as explained in the instruction manual.⁴

⁴ The manual states each level contains six stars, but they also contain a seventh, hidden star.

SM64's PRSes Arguing Toward Rules

The function of PRSes is similar to argumentative mechanisms in games Bogost (2007) discusses, albeit concerning rules of games themselves rather than material world implications. Bogost explains argumentative mechanisms in games via analogue to “Aristotelian enthymeme. The enthymeme is a technique in which the proposition in a syllogism is omitted... the listener (in the case of oratory) is expected to fill in the missing proposition and complete the claim... interactivity can produce an effective procedural enthymeme,” (p. 43). Games achieve a similar sort of enthymematic engagement with players during gameplay, as players agree to arguments about game rules while simultaneously actualizing them through interaction.

In *SM64*, there exist numerous examples of this sort of argumentation toward the player, with many of the game’s PRSes arguing toward its star-based progression system specifically. Players are, of course, not to access new areas in the castle without the required number of stars, and the game constantly argues toward the importance of stars using one simple object - doors. In terms of checkpointing, doors are clearly environmental. They are easily visible in the hub, and passing through those which are locked (Figures 2 and 3) visually demonstrates progression. However, because of the way that doors are unlocked, they are connected to UI checkpoints too. If a player’s star count is insufficient, they will most likely become aware that they cannot open some doors without raising that count, cultivating a connection between the UI and environment. Doors serve as PRSes due to the multitude of implied and concrete systems they are attached to, making numerous arguments as a result, but one of those arguments is strongly tied to progress. This argument assumes the form of an enthymeme which is further detailed later in the chapter, and the incomplete rhetorical proposition can be rendered as such:

“Stars are how you open locked doors and make progress.”

The first encounter with the PRS of the door is also the most illuminating in all of *SM64*, because it serves as the player’s mechanical introduction to the game’s rules and progression. This occurs when the player initially enters the first floor of the castle - the first area of the hub. Throughout the first floor, there are many doors, most are locked to a player with zero stars. Approaching locked doors yields a message which informs the player that they need more stars. Meanwhile, among the few unlocked doors, there is only one leading to a star-containing level,⁵ one universally considered the first level in the game - Bob-Omb Battlefield.

The locked doors on the first floor create an enthymeme players are expected to take up in a specific way, their unstated conclusion being highly evident in *SM64*’s environment design. They plainly state that there are limitations on where a player can go in the game at a given time, yet in doing this, they tell the player they should be seeking ways to get around those limitations. Playing a single-player game means establishing a minimum stasis between player and software indicating that the game can be beaten, meaning that a limitation should generally be passable. *SM64* players who explore for long enough will eventually find their way to means of passage, unsurprisingly taking the form of the first level-containing door leading to Bob-Omb Battlefield. Recognizing the entry of levels accessible by way of one’s star count as a means of progression, players will likely accept the game’s argument and view this as how it is meant to be played.

This is the enthymematic argument that the game makes during its introductory moments, and it is one the game continues to make every time the player encounters an unlockable door. Yet

⁵ The other doors lead to a hallway which has two main points of interest. One is the staircase to area 2, which requires a star count of eight to be opened. The other one is a door which leads into a courtyard. The courtyard is unremarkable at first, but the entrance to the level Big Boo’s Haunt appears here when a player raises their star count to twelve.

with the status of doors as a PRS, one cannot forget the necessity for players to take it up, alongside the variety of different paths players may take before they arrive at standard uptake. The functionality of the doors' argument, which tries to keep players increasing their star count, isn't intrinsically linked to how players might take it up when first experiencing the enthymeme. When the argument is understood by the player, a choice must be made about if they'll follow it. Players who take up the door PRS in the way that it was specifically meant for them to take it up, proceeding to play through the game letting their star count determine their level of progression, will be playing by standard uptake, perceiving the environment according to the game's rules. Few would doubt the majority of *SM64* players adhere to this sort of uptake and accept its PRSes - yet with PRSes "in action" having been shown, what does standard uptake of them look like?

Standard Uptake of *SM64*'s PRSes

Observing standard uptake in action is fairly easy thanks to the existence of YouTube. Videos featuring gameplay and commentary show how players tend to take up rhetoric in games, and the majority of these sorts of videos showcase players utilizing standard uptake as they play. In particular, "Let's Play" videos, or LPs, are useful. This YouTube-based genre has grown very popular since the turn of the 2010s, and it consists of a relatively easy-to-understand formula. Players record footage of a game as they play through it and offer commentary over the footage. To get an authentic sense of "play," most LPers prefer to commentate while they are recording, with their reactions being created in the moment of interaction with the game.

Because of this, LP content shows uptake of various players as they experience games. While it is true that LPers constantly have to split their attention between commentary and play, they tend to adopt a relaxed and casual style of delivery, opening a window into their thoughts. *SM64* LPs are interesting in this regard. Commentary and play in them can often vary wildly, since

the game has existed for so long that some have played it repeatedly since it first came out, while others have recorded “blind” LPs in which they play through the game for the first time. Blind LPs have a noticeably higher chance of demonstrating standard uptakes as they develop, both in play, as players use trial and error to figure out the game’s arguments, and commentary, since they will frequently verbally announce what they have taken up from the game’s PRSes.

Blind LPs are useful to analyze because seeing how people take up PRSes for the first time helps to further clear up how the enthymematic mechanism behind PRS uptake works. YouTuber Flapper (2018) provides a perfect example of this with his blind playthrough of *SM64*. Upon entering Peach’s Castle, Flapper spends several minutes wandering around the first floor. Eventually, he approaches a door to a level. However, it is not the door to Bob-Omb Battlefield, and all other level-containing doors need at least one star to open. Flapper quickly realizes this; receiving a message from the door, he says, “I’m assuming you need a star... to go through.” Interestingly, Flapper is playing the Japanese version of *SM64* and can’t read Japanese himself. Despite not being able to actually know the door’s message, he takes up its argument instantly. He attempts one other door that gives him the same message before he narrows his search radius, eventually bringing him to the door to Bob-Omb Battlefield. He proceeds to finish the first objective in this level, gaining a star and satisfying *SM64*’s progression system (Figure 6).



Super Mario 64 - 120 Star Run - *Blind Playthrough*

409 views

30 1 SHARE SAVE ...



Flapper

Streamed live on Jul 15, 2018

SUBSCRIBE 2K

Checking out Super Mario 64 for the first time tonight. Let's see how well I do playing a Japanese copy.

Figure 6. Content creator Flapper realizes what *SM64* is arguing he should do. After some poking around in area 1, he locates the door to Bob-Omb Battlefield.

In this instance, the PRS of the door has fulfilled its purpose in convincing a new player to play *SM64* by its intended rules. Flapper knew the answer to the rhetorical proposition he was presented with the moment the game imposed a limitation on his exploration of its environment. It is possible that with his interest in retro gaming (which he stated in his channel description), Flapper already possessed at least some slight awareness *SM64*'s rules and progression systems. Yet knowing the rules of a game does not ensure a player will follow them; the game must make its rule adherence argument known through limitations it imposes in the structure of its world. Flapper encountered and accepted this argument, leading him to demonstrate standard uptake, playing the game in the way it is meant to be played.

Demonstrations of standard uptake can also be quite interesting with experienced players. Even more interesting is the result of players of varying experience commentating concurrently,

particularly when the experienced have a better grasp on standard uptake than the inexperienced. The video *TEENS PLAY SUPER MARIO 64 - Part 1* (FBE Live, 2018) shows a group of three with that dynamic, with one among them being obviously more experienced than the other two. Since *SM64* is single-player, only one among the group is able to play while the other two watch. The most experienced player ends up in the observing category while one of the less-experienced players assumes the controls. During a scene extremely similar to the one experienced by Flapper, the player behind the controller wanders the first floor of the castle. The more experienced player says that he knows what to do, and after a bit of back-and-forth, the less-experienced player accepts his directions and finds his way to Bob-Omb Battlefield.

The players' conversation could not have occurred without the mutually-understood premise that *SM64* imposes certain limitations on player actions. The less-experienced player, before even getting a proper objective, knew he was expected to follow a specific set of rules. The more-experienced player, an apparent "expert" on the game, relayed the specifics in question because he understood what his less-experienced fellow was seeking. Both players accepted there was a way that the game was meant to be played without question. Conversations like these are what surround standard uptakes - an understanding of what the game means the player to do with its tools and how it enthymematically argues the uses of those tools. Players are so used to encountering these mechanisms in games that they actively seek them out, anticipating rules will be argued to them likely based on past experiences. This, unsurprisingly, leaves another question - why do players follow standard uptakes so readily instead of trying to get around them?

Cheating Versus Standard Uptake

Cheating is the most obvious way that one would attempt to work around a game's rules, and thus, play in such a way that they would take up PRSes while not accepting their arguments.

One who maxed out their star counter in *SM64* without actually collecting a single one of them, or advanced to a certain room in the game lacking the necessary stars to be there by the rules, would be ignoring certain rules and thus seemingly not taking up the game the standard way. What motivates the majority of players to not cheat and instead fall under standard uptake, then?

This question doesn't seem hard to answer if viewed from a certain material perspective. That is, assuming that a *SM64* player has access to nothing other than the console and the game. In this situation, *SM64* does not offer any built-in cheat codes like other games from its era. Considering this, there does not appear to be a way to act on anything other than standard uptake. This will be proven untrue in chapter 3, but for the moment, it is safe to assume the average player will not have access to means of cheating their way past the intended way of playing. However, while this certainly can affect players' views of rule rigidity, it is not the only option, and even when the game came out there was potential for branching off from standard uptake.

Yet in the absence of cheat codes, cheat devices have existed for the N64 since launch. These devices are plugged into consoles and allow for game code to be altered, granting players the ability to cheat where no such opportunities might have existed previously (Nilsen, 2010). The option to use these devices has existed since the beginning of home console video gaming, though there may be arguments against considering them as enabling non-standard PRS uptake. Namely, that they are not accessible to all players due to distribution, cost, and other concerns, and that without them, cheating in *SM64* still appears impossible for the majority of players.

The apparent inability for the majority of players to challenge *SM64*'s arguments is clear. This has given players reason to consider rules not as rhetoric, but as unbreakable, undeniable facts of the game's world. Chris DeLeon, in *Rules in Computer Games Compared to Rules in Traditional Games* (2013), argues toward this notion by comparing two theoretical mazes - one of

tape and one of concrete. The tape maze is followed by convention, since it is possible for anyone to step over the tape. The concrete maze is followed thanks to the rules of materiality, since its walls cannot be bypassed. DeLeon argues games are more similar to the concrete maze, as, like the door PRSes in *SM64*, they do not appear to offer much in the way of choice.

DeLeon's argument holds water when considering games like *SM64* in a historic sense, which assumes a player interacting with the game had the aforementioned material limitations. However, much has changed since 1996, and game texts have become increasingly open-source, to the point the notion of games as concrete mazes, which might have been stronger in the past, is gradually fading away in the modern age of interactive software. One area this is apparent, particularly for more tech-savvy gamers, is emulation.

Emulation is a popular practice for playing games on consoles that are out of production, with the long-discontinued N64 certainly being on that list. It entails recreating the operational environment of one device using another - usually a personal computer strong enough to run it. Creators of emulators are given the unique opportunity to put their own spin on the software, including enabling cheats in games which, in their original, untampered-with forms, had none. Project64 (hotnintendo, 2009) makes it incredibly easy to use cheats in a game like *SM64*, meaning that anyone who decides not to take up the game's arguments and is playing on an emulator platform has all manner of options for alternative play.

Barriers from the past which might have prevented players from taking up PRSes in non-standard ways are thus no longer in place, at least to the same degree. However, despite this, DeLeon's argument should still be acknowledged with respect to a very particular circumstance: that of material and social limitations further restricting possibility for alternative PRS uptakes. Not all players will have access to emulators, and by extension, not all players will have access,

or *want* to have access, to the wider community of *SM64* players who often share their uptakes. Such conditions would preclude players from adhering to, or perceiving the game through use of, any uptakes other than standard. Their perception would likely be such that the game's rules do seem to be inviolable facts of programming.

Of course, this scenario once again seems more likely to be historical in the current era of games and the vast information networks connecting them online. *SM64* is not an online game, but anyone with an interest in it can discover YouTube videos and other stores of information which can reveal non-standard uptakes of its PRSes. Even if they may not have the ability to actually play according to those uptakes, the simple knowledge they exist is important by itself. The idea of games as a concrete maze has become flimsy precisely because the majority of players will at least be aware that *SM64* doesn't need to be played exactly according to its rules. In addition, even a player with no means of cheating or accessing the information circulating the internet regarding *SM64* may still ask themselves whether there are other ways they could play. Cheat devices could not have been created if they had not been preceded by the idea that there may be ways of playing games which do not strictly adhere to their rules. Though most current *SM64* players will perceive and play according to standard uptake, it is unlikely that they would claim that their reason for adhering to the rules is because they have no options.

And regarding the topics of "most players," standard uptake, and online communication, one need only look at video makers and livestreamers who play *SM64* via emulation to see that most of them choose to follow the rules even when several other options exist at their fingertips. TTDLX (2011), for example, plays through the entire game on an emulator collecting every star, with the intent of demonstrating what it looks like for a player to overcome all of its challenges. Vinny of streaming channel Vinesauce (2017) also plays through the whole game on an emulator,

and while he doesn't collect every star, comments on his stream are interesting in their own right, as they demonstrate the opinions players can have relating to how games are meant to be played. Gameplay videos like these show that outside of being forced into following the rules of *SM64*, players have other reasons to adhere to rules even when they could easily choose to break them, suggesting that motivations for standard uptake are more complex than they seem.

Player Motivations for Adhering to Standard Uptake

Explaining these reasons means delving into motivations for following standard uptake. Such motivations are likely innumerable for each individual player and their reason for playing. However, it is possible to narrow motivations into at least two primaries, both linked to rhetoric. In general, rhetoric connects to player motivations due to its nature and relation to persuasion, and persuasion is heavily integrated into the motivation to follow the rules of games in general. With an analog game like chess, there is no fundamental, universal law which dictates that players must interact with pieces on the board in a specific way. There must be an agreement between both players that leads to mutual acceptance of the same rules. Without the mutually agreed-upon premise that there are limitations on player actions, there is no game.

This might sound similar to a concept discussed in section 4 - that of players and games, along with players and other players, needing to mutually accept that rules will be part of *SM64*. This agreement to follow the same rules between software and playerbase is procedural rhetoric, reinforcing qualities video games have that make them similar yet different to analog games. There needs to be acceptance of the argument that a game should be played some specific way, like in analog games; yet unlike with analog games, video games communicate their rules more or less autonomously, teaching their rules through procedural rhetorical systems. When players accept the arguments posed by these systems, they will uphold them when playing around others, leading to

standard uptakes of PRSes in games like *SM64* becoming commonplace. Two main motivations for adhering to standard uptakes emerge from the game's argument being understood via simultaneously accepting and enacting that argument; communal, relating to how people perceive others playing the game, and satisfaction, relating to what one wants from a game.

Because of this, one of the motivations behind standard uptake adherence is communal. Players observing other players will most likely anticipate that they'll adhere to standard uptake, since without that uptake the meaning of playing the game can be disrupted in the eyes of many. Alternative uptake of a game's PRSes would likely be viewed as a disruption of the rule accords. This is importantly connected to LP and streaming content, in play is scrutinized by audience.

While Vinny of the channel Vinesauce does not disrupt any rules himself, his audience displayed frustration at the beginning of his stream (2017) of *SM64*. The issue they took was with Vinny's choice to play the game by emulating its original N64 version instead of the 2005 remake of the game released for Nintendo DS. Some fans believe the DS version is definitive,⁶ and it may be within their belief that it contains the more relevant set of rules and PRSes, considering that the developers would've seemingly had more time to polish them. This, in turn, would make the remake more worth playing, since its rules are more officially up-to-date.

Vinny nonetheless pressed on with the original, and his viewers eventually quieted down, but their reaction is still worth noting. Fans of video games are very particular about how games are meant to be played, and there is evidence to suggest they tend to advocate standard uptake. One example can be seen when media outlet IGN (2018) posted a speedrun of 2016 game *Doom*. *Doom* is a reimagining of the classic 1993 first-person shooter, and much like in the original,

⁶ Some fans believe the DS version to be superior because it includes more content. Others have less favorable views of it because it removed many glitches that made the N64 version popular for speedruns - a topic further explored in chapter 3.

progression is achieved by moving through chambers filled with demons and slaying them, which unlocks more chambers filled with more demons that serve to keep the carnage going. This can easily be described as how *Doom* is meant to be played.

However, the speedrunner almost completely ignores the progression rules of the game. Since their goal is to beat the game as quickly as possible, they make use of various glitches, including many which allow them to pass through walls and skip large sections of the game. Likely because the speedrunner's video was uploaded to a channel with a mainstream following, there was some backlash when the video was initially uploaded, with a like/dislike ratio of 6:1⁷. Some comments decried the player of cheating, while others claimed that he was not cheating, since he was not using any means that exist outside the game.

Whether speedrunning technically construes "cheating" will be discussed in chapter 3. For now, it is important to note the backlash here resulted from perceived injustice to the game. The game offered the player an argument on how they should be playing it; the player refused, choosing to play their own way instead. This disrupts the agreement between player and game, and arguably more importantly in a social media space, the agreement between player and player. Some viewers of the speedrun were expecting the player to quickly play through the game using a standard uptake lens and were frustrated to have their expectation subverted.

While speedrunning and cheating are different, they both constitute non-standard uptakes. As can clearly be seen in the comments of the *Doom* speedrun and on Vinny's *SM64* stream, audiences for this content have specific expectations for how games are meant to be played. It's not dissimilar to how an audience watching a game of basketball would expect to see players

⁷ YouTube allows users to "like" or "dislike" a video. Videos that are generally accepted have like/dislike ratios skewed heavily towards likes. Any video with a significantly high number of dislikes suggests that there was friction between audience and content.

adhere to certain rules - every player and audience member must believe those rules as absolute. Thus, Vinny's choice to follow *SM64*'s rules despite having access to cheats was, on some level, at least partially influenced by the expectation to play the game as it is meant to be played.

Of course, to say that players of *SM64* play according to standard uptakes purely because they are afraid of being criticized for playing the game incorrectly would be oversimplifying. External motivations can certainly have an effect on player behavior; comparing responses of players who were studied under varied conditions seemingly suggests as much⁸ (Mackey, 2007). Yet internal motivations can be just as important as the external motives discussed previously.

What inspires someone to place limitations on themselves in order to tackle a challenge? The exact answer to this question is unknown, though Bogost points it out as endemic to play. Within the first few pages of his book *Play Anything* (2016), he talks about a moment in which he and his young daughter were walking through a crowded mall. Apparently bored, she came up with a simple game based on the tiled floor beneath her, attempting to land her footfalls only on the tiles themselves while avoiding stepping on the grout-filled cracks in-between. This is a game that every child probably challenges themselves to at some point, and it comes from the child's recognition that to "play," there must be limits. As Bogost explains;

"This paradox of play—the idea that fun arises from limiting freedoms rather than enhancing them—isn't only true of board games or card games or playground games or video games. It can be found in any kind of material whatsoever." (p. 66-67)

Of course, if limitation is essential to the enjoyment of video games, then motivation for following adhering to standard uptakes and avoiding the use of cheats starts to make more sense. Playing a video game adhering to standard uptake means taking up the PRSes as they are meant to

⁸ Mackey's (2007) study recorded teens aged 14-17 playing the same game with different partners, yielding different results in terms of whether they focused on the game's mechanics or story.

be taken up, and in most cases, those PRSes are arguing to the player that their actions are limited in some way. Being capable of finding solutions to the problems that those limitations create is where the joy of playing a video game truly shines through.

The nature of limitation is also the reason that most academic studies of game rules (particularly those citing Gee) tend to adopt a lens of learning. Learning and skill development are obviously major parts of playing video games. The learning lens becomes particularly applicable when discussing “epiphany” moments - the sudden sparks of realization which characterize enthymematic interactions with PRSes. Epiphanies cannot occur without *aporia*, which represents the feeling of being “stuck.” (Sezen and Isikoglu, 2007). The importance of feeling as though one has overcome a challenge is very important to the appeal of video games, as many studies on the concept of “game flow” have attested (Sweester, 2017). Meanwhile, as Gee (2003) has pointed out, players also tend to feel more accomplished when they feel as though they have learned some aspect of the game’s systems and used those systems to their advantage in figuring out how to progress, not unlike the feeling of satisfaction one gets from learning a challenging concept in class. The educational and “flow” based models have great merit, especially as some of those concepts are necessarily adapted to a rhetorical lens.

Accepting PRS arguments and learning the rules of the game as it is meant to be played is the premise of standard uptake. Assuming standard uptake is not only the basic form of uptake, but also the “standard” in the sense of what people expect to follow, this aspect of rhetorical rule enforcement for video games could easily be fit into flow and educational lenses. Gee (2003), Sweester (2017), and others need only be assumed as describing standard uptake in regards to flow and learning. Comprehending a good and challenging game through standard uptake is action which requires focus and close reading of the game’s PRSes, and although players who are aware

of uptakes other than standard have a choice in whether they adhere to this uptake, it is certainly attractive for the form of satisfaction it brings.

Such satisfaction could very well be lost if a player experienced a game using cheats. Sezen and Isikoglu (2007) explain this effectively when they describe how a player would feel when playing a game using “god mode,” or invincibility. “Playing in special modes-such as God mode-prevents the pleasure derived from game play. One of the basic components of game concept is risk, i.e. the possibility of losing.” Sezen and Isikoglu also describe the moment of “aporia” that comes from working through limitations and discovering an aspect of how a game is meant to work. This is part of why Sezen and Isokoglu acknowledge that using strategies guides could be considered cheating, since they seemingly rob the player of aporia (p. 3-7). However, the existence of hints in game manuals, like what can be seen in Figure 5, may complicate this idea since game makers clearly are not opposed to communicating their rules using methods outside the game itself.

Regardless, the nature of satisfaction shines through in LPs and livestreams of *SM64*, where players appear to adhere to standard uptake because the limitations it poses create a more rewarding experience. Even with immediate access to means of cheating, it is often more enjoyable to act within limitations than to allow oneself complete freedom. As mentioned above, Bogost himself argued that complete freedom eliminates the enjoyment that can be gained from playing a game. Blind LPers are likely to desire playing the game as it is meant to be played, taking up door PRSes as sturdy progression barriers, because the experience of parsing the game and learning its rules is more enjoyable and rewarding than simply cheating one’s way through it. More experienced LPers and streamers, meanwhile, are likely presenting the game this way both because they are expected to, and because they recognize that it is usually considered more interesting to watch a game be played in the way it is meant to be.

CHAPTER III: SPEEDRUN UPTAKE

Chapter Introduction - Speedrun Uptake

This uptake lens (UL) gets its name from the practice that it is based on - speedrunning. Speedrunning is “playing a game with... intent of completing it as fast as possible” (SRL, 2019). In the interest of speedy completion times, speedrun uptake allows procedural rhetorical systems (PRSeS) to be taken up in ways which can (and often do) ignore or reinterpret their arguments. The reason for these PRS reinterpretations has to do with special techniques used during runs⁹, including some of which are commonly referred to as “skips.” Skips, as their name implies, allow for large portions of the game to be circumvented. This is a cause for PRS reinterpretation; since games’ intended progression systems are often thrown into chaos by these sorts of skips, speedrunning often involves playing the game in ways it wasn’t meant to be played.

However, pulling off the techniques required to accomplish skips is notoriously difficult. Even being able to take up the game’s PRSeS in a way that allows for this loose reinterpretation, the “uptake” part of speedrun uptake is withheld behind a study wall of specialized knowledge. This knowledge can be acquired fairly easily thanks to the internet, but its relative obscurity compared to the way the game is meant to be played prevents most players from experiencing it. Speedrun uptake is thus less common than the last chapter’s UT, standard uptake.

However, despite all of this, speedrun uptake is nonetheless connected to standard uptake. To “finish” a game as quickly as possible means reaching the ending intended by the developers, which means PRSeS connected to progression must be followed at least to a degree (SDA, 2019). Even if speedrunners do skip large portions of the game to cut down on their overall playtime, the ending they see is the same as the one seen by those who play the game in the intended way.

⁹ Short for “speedrun,” which is an individual attempt at playing through a game as quickly as possible. “Running” refers to the act of speedrunning, while “runner” is short for “speedrunner.”

There is obviously a great deal to unpack when it comes to speedrun uptake and PRSes. Thus, this chapter focuses specifically on *Super Mario 64 (SM64)* and its speedrun community, analyzing it from several different angles to paint a clearer picture of speedrun uptake as a whole. Like chapter 2, this chapter starts with a top-down approach which establishes several rules. Firstly, certain rules which are in effect within both standard and speedrun uptake are discussed, those rules having to do with how the game's progression leads a player to eventually finish it. This means introducing the concept of the win-state, along with a few additional checkpoints, which will be further detailed in section 2.

The second batch of rules on the "top" level actually concerns speedrunning specifically. Despite appearing to be all about breaking limits, speedrun uptake in general adheres to rules, though unlike with standard uptake, these rules cannot be not found anywhere in the game itself. They are found on speedrunning websites, which describe how the game is meant to be played from a runner's perspective, thus informing aspects of speedrun uptake.

One possible complication is the existence of multiple categories for *SM64* speedruns, which include 120 Star, 70 Star, 16 Star, 1 Star, and 0 Star, each one providing different rules. These categories will be explained in greater detail in section 3, but for now it is important to note that for keeping things fairly streamlined, only runs in the 0 Star category will be observed. With this category and its rules established, speedrunners' relation to PRSes can be discussed. Similar to before, YouTube videos are the medium of choice for detailing how those utilizing speedrun uptake navigate PRSes compared to those utilizing standard uptake. However, due to the complexity of speedrunning, only one video is emphasized this time, though like before, there is one PRS interaction that takes center-stage, as it is linked with the game's biggest skip.

The Checkpoints and Ultimate Win-State of *SM64*

A game's win-state is, in a single-player game like *SM64*, a goal determining completion (Cruz-Cunha, 2012). Win-states range from short-term goals, like the level-ending flag in *Super Mario Bros. (SMB)*, to the ultimate, long-term goal of a game, such as clearing the final level in *SMB*. Regardless, they are the goals which rules and progression systems push players toward. Oftentimes, win-states can also be viewed as progression checkpoints. However, in most games, the "ultimate goal" win-state is not a checkpoint since no progression typically exists past it¹⁰. Rather, the ultimate win-state is where the game definitively concludes.

SM64's win-states are actually quite similar to those of its two-dimensional antecedent, albeit recontextualized in a way which better fits its explorable three-dimensional environment. *SMB* tasks its player with progressing through eight different worlds, each containing four levels. In the fourth level, the player must fight the boss, Bowser, in order to advance to the next world. This continues until the player reaches world eight and engages in one final battle with Bowser. After this, the game ends, making the final Bowser fight in *SMB* the game's ultimate win-state, while all the other Bowser fights before it can be considered checkpoints.

SM64 also emphasizes boss fights against Bowser as primary markers for progression. Naturally, the way it structures these battles is slightly more complicated than the original *SMB*. As was discussed in chapter 2, players make progress by moving through four areas in the hub. However, while it may seem like the win-state is raising the star count high enough to advance, this is actually a requirement that the player must meet before they can find the actual win-state. This is because there are two door types linked to progression - big star doors and keyhole doors.

¹⁰ This notion is complicated by the existence of games with a "new game plus" feature, which allows players to start the game over and play again to access more content. However, *SM64* and *SMB* contain no such feature. As such, their endings are definitive.

Big star doors (Figure 7) are opened by having a star count matching their requirement. These doors lead not to the next area, but to Bowser levels. Unlike most of the game's levels, which are open and allow for exploration to find stars, Bowser levels are a more focused affair; essentially, linear gauntlets which contain some of the most difficult challenges in the game. After the player gets through these challenges, they meet Bowser at the end of the level, proceeding to engage in a boss fight, similar to *SMB*. After Bowser is defeated, he drops a key. This key is used on a keyhole door, which allow the player to progress to the next area.

SM64's Bowser levels serve the same progression purpose as *SMB*'s end-of-world levels. They are the goal that the player is pushing toward, with all PRSes and progression systems visible in standard uptake urging the player to raise their star count high enough to enter them, overcome the challenges, and ultimately defeat Bowser. This formula is not entirely consistent, however, and it is for that reason that it is helpful to look at how each area transition is handled, which gives perspective on how the game's progression loop leads to its ultimate win-state.

The first instance of the Bowser level win-state occurs, unsurprisingly, at area 1's end. Once the player raises their star count to eight they can pass through the sliding doors which lead into the first Bowser level, Bowser in the Dark World (referred to as "Dark World" for brevity). Clearing this level grants the key needed to unlock the keyhole door (Figure 8) leading to area 2, also known as the basement. There, the player must visit levels to raise their star count to thirty, gaining access to big star door which leads to the second Bowser level, Bowser in the Fire Sea ("Fire Sea" for brevity). This grants the key which allows them to advance from area 2 to area 3, also known as the second floor. This is the point where the progression system becomes slightly (though not confusingly) inconsistent.



Figure 7. Big star door of area 1. The star count must be at least eight to open it.



Figure 8. Keyhole door leading to area 2. Opened with the key from Dark World.

Rather than open another big star door that would permit access to another Bowser level, the sliding door at the end of area 3 permits access to area 4. This is the one point in the game where the win-state actually is to raise one's star count high enough - in this case to fifty stars - in order to achieve progression.

However, this inconsistency is isolated, as area four sees players raising their star count to seventy to access the final Bowser level. Even so, for dramatic and climatic effect, players do not pass through a big star door to reach the final Bowser level, but rather, the endless staircase. Befitting its name, the staircase is supposed to loop infinitely if the player attempts to climb it without seventy stars. Once the star count is raised to seventy, the staircase ceases to be endless, allowing access to the final level, Bowser in the Sky (or simply “Sky”). Upon defeating Bowser, the player gets one final star, triggering the ending and serving as the ultimate win-state.

With Bowser the levels outlined, *SM64*’s rules and progression get a new layer of clarity. Players are intended to enter an area, visit levels, collect stars, pass through big star doors, complete Bowser levels, and advance to a new area, where the cycle repeats until the game ends. *SM64*’s ultimate win-state thus becomes very easy to discern even to a completely novice player, and as such, there is little question among players regarding what constitutes “beating” the game. If one were to claim that they could complete *SM64* faster than anyone else in the entire world, what they’d mean from a mechanical standpoint is that they could trigger the win-state quickly.

Acceptance of win-state as grounds for beating the game is needed for it to hold meaning. This is true whether the game is viewed through the lens of standard uptake or speedrun uptake. However, speedrun uptake has the unique quality of being able to take up PRS arguments in ways which are not possible in standard, as some aspects of the game’s rules (such as the need to collect stars) result in slower playtimes. Yet in addition to still following the ultimate win-state, speedrun uptake also follows many rules of its own. These rules are unique because they are not imposed by the game itself, but by speedrunners, and their choice to follow such rules reinforces the notion of game rules being rhetorical in construction.

Speedrun Categories and Rulesets

The rules which runners follow also serve as the rules which govern speedrun uptake. Runners used to keep track of their rules with informal means when speedrunning was (more of) a niche practice. In recent years, however, speedrunning has gained more and more practitioners, leading to an increased need for the rules to be recorded using online databases and web pages. Among those sources is the well-known Speed Demos Archive (SDA). The facts and questions (FAQ) page on SDA establishes several rules which are generally adhered to by runners.

Relevant to this chapter, and the one before, is SDA's statement on the matter of cheating. Use of cheat codes and any devices external to the game hardware are banned in speedrunning. In addition, runs are (at least according to SDA) generally only accepted on original hardware, meaning that emulators are typically not permitted (2014). Ultimately, SDA seems to heavily emphasize the "purity" of runs, meaning that the cheat codes, devices, emulators, and other means of getting around game rules mentioned in chapter 2 are not allowed. Runs are generally expected to be performed using means readily available in the game itself - nothing more, nothing less.

This does not mean that there are no exceptions, however. Runners have raised material concerns relating to the availability of consoles like the Nintendo 64, which are no longer made. This led to the creation of categories for runs on emulators and runs on original hardware. "Speedrunning categories" generally refer to the conditions under which runs are performed, which can include the hardware platform. This is to ensure that results of runs under different conditions are not conflated with each other due to the number of variables at play.

Of course, categories can refer to more than just the machine that a run is performed on. They can also represent the rules and limitations which runners choose to abide by. Speedrun.com, a newer and highly popular resource for speedrun information, demonstrates this. The site contains

a page for seemingly every speedrun-friendly game, *SM64* definitely included. Of particular note are the categories indicated on *SM64*'s page, which are rather unique in terms of the limitations they prescribe for certain types of runs.

An overview of *SM64*'s speedrun.com page would most likely help to elaborate on this. Like other pages on the site, it contains a leaderboard displaying the fastest completion times. And yet, the visible leaderboard is actually one of many, each representing not just one category, but a combination of two dictated by the category tabs. *SM64*'s page has two “layers” of tabs. The “lower” layer relates to the hardware platform, previously mentioned as being a point of contention among runners regarding whether a “pure” run requires playing on original hardware. Speedrun.com tries to resolve this issue by providing categories for original hardware (N64), emulators (EMU), and virtual console (VC). The last one (VC) doesn't require much elaboration, as it is an emulation service Nintendo provides on their own consoles (Speedrun.com, 2019).

Yet in addition to tabs relating to platform, there is also the “upper” layer (Figure 9). These tabs represent categories related to the rules that a runner must follow during their run. These rule-focused categories exist for every game on the site, since unlike with standard uptake, there is no definitive way any game is “meant to be” run. Categories are usually broken up by how runners have chosen to take up (or ignore) PRS arguments relating to rules and progression.

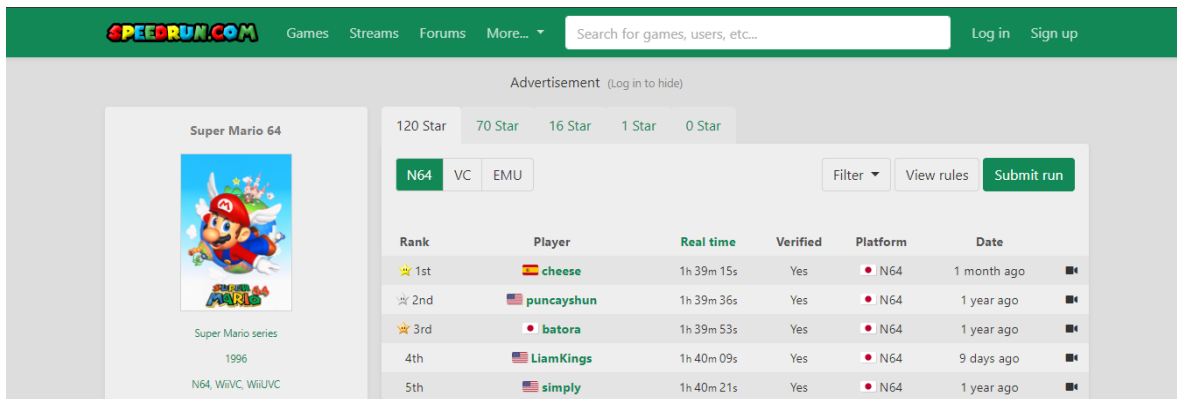


Figure 9. *SM64*'s page on speedrun.com with the 120 Star category tab open.

SM64's categories are no different in this regard, though they prove somewhat unique. Unlike many other games' categories, *SM64*'s focus on a specific element of progression - stars. The top layer of category tabs are therefore labeled 0 Star, 1 Star, 16 Star, 70 Star, and 120 Star. Evident in each category is the star count that the player must have when they finish the game. This fact alone holds all sorts of implications for how speedrun uptake views the game's PRSes. With a category like "0 Star," speedrun uptake is obviously deviating a fair bit from standard, heavily suggesting that runners only follow aspects of the game's rules out of their own agency. Furthermore, the categories of *SM64* speedrunning differ so much in the rules they prescribe and how they are executed by players that in some ways, each could be considered its own game.

The categories are easiest to explain by simply starting at the top and working downward. *SM64* contains a total of one hundred and twenty stars in its various levels for players to gather, and as such, 120 Star essentially means doing everything in the game in the fastest way possible. This results in hour-and-a-half-plus playtimes - incredibly lengthy by *SM64* speedrun standards. For this reason, glitches which let runners skip sections of the game (known simply as "skips") are fully allowed in 120 Star, as they cut down on considerably on playtime.

However, use of such glitches is not always permitted. While *SM64*'s categories may look like they only concern the star count a player has when they finish the game, one of them - 70 Star - also prohibits the use of certain glitches which allow for large skips (maxaura, 2017). This is due to 70 Star's conception - the category represents the minimum number of stars required to beat *SM64* normally. Some have described it as the "pure" way of running *SM64*; since the category uses few glitches, it could be described as adhering closely to standard uptake.

If *SM64*'s intended rules continue to be viewed through the lens of standard uptake, adherence to them declines when discussing categories with star requirements lower than 70 Star.

This is by virtue of lower categories going below the minimum the game is intended to require, and because of this, the game's normal progression and rule systems are thrown out the window. Speedrunners are not required to enter levels, collect stars, and unlock Bowser levels to make their way to the game's ultimate win-state. Rather, they place those limitations on themselves for the sake of redefining the type of challenge they are tackling.

For brevity, the 16 and 1 Star categories are glossed over. They have unique qualities and certainly prove that *SM64* runners have a choice in how they take up progression-related PRSes. However, this demonstration of PRS agency is accomplished by all the sub-70 Star categories, meaning that it is only truly necessary to examine one of them. With this in mind, the sub-70 Star category demanding attention should be the one with the most unique implications for PRSes, strategies/glitches, and concrete examples of speedrun uptake in action. Per these parameters, this chapter explores a category that shows how much speedrun uptake can bend rules to the wildest possible extent, which for *SM64* speedrunning the 0 Star category.

As its name implies, this category simply requires players to complete the game with a star count of zero. Of course, calling zero stars a "requirement" would be somewhat misleading, at least from a certain perspective. It has long been understood that it is possible to clear *SM64* with zero stars, and for the most skilled of speedrunners, the other categories could alternatively be described as different star count requirements or as different levels of self-imposed limitation. 0 Star places the fewest limitations on players, though they still must follow some basic rules. Like in every other category, there is one main PRS that needs to be very clearly adhered to and is the same as it would be in a playthrough adhering to standard uptake. That PRS is, of course, the game's ultimate win-state: the final battle against Bowser and acquisition of the final star.

Everything else, however, is essentially fair game for runners in the 0 Star category. Because anything goes in the 0 Star category, it is also necessarily the category which contains the most frequent use of glitches, to the point that most 0 Star runs are primarily glitch-based. This makes 0 Star runs quite interesting when done on original hardware (the N64 category), since they prove that with a combination of speedrun uptake and a particular mechanical skillset, it is possible to have agency in PRS uptake even without the use of cheats or emulation.

Of course, to understand why 0 Star runs are able to deviate so much from standard play, it is very helpful to acknowledge the glitches which enable this kind of play in the first place. Such glitches are responsible for enabling skips, which are, of course, core to the 0 star category. This makes these glitches and the skips they enable crucial to speedrun uptake, because they represent the possibility of agency in PRS uptake from the puzzle-like angle of speedrunning. One glitch, and its associated skips, is particularly useful for understanding how 0 Star runs achieve incredibly unique PRS uptake even on original hardware.

The BLJ - *SM64*'s Biggest Skip-Enabling Glitch

The backwards long jump - or BLJ - is not a move players are intended to use in *SM64*. Despite this, it is a skill well-known by *SM64* players of all sorts, with detailed documentation on sites like speedrun.com explaining how to perform it (Alaris_Villain, 2017). It's a skill that is considered by some to be essential for running any category except for 70 Star, and like many speedrunning techniques, the key to performing it isn't found in the game itself, but within the speedrun community.

However, understanding the BLJ is best done by first understanding a two basic points about how *SM64* works as a piece of video game software - speed and collision detection. Regarding collision detection, 3D games are use flat planes called collision meshes to calculate

where one object makes contact with another. Collision meshes are easy to imagine in older 3D games like *SM64* thanks to their “blocky” appearance, which stems from the N64’s hardware only allowing for low polygonal detail. The flat surfaces inside Peach’s Castle can be imagined both in terms of their visible textures and in terms of the underlying collisions.

Figure 10 shows a wire frame view of *SM64*’s castle foyer, demonstrating how the game’s geometry is made up of flat planes dictated by collision meshes. Collisions push back against all objects, including Mario, that make contact with them. They are the reason that the player cannot ordinarily pass through walls and other objects. “Ordinarily,” because one of the main functions of the BLJ is passing through collision planes, resulting in what is commonly referred to as “clipping” in the 3D graphic design discourse. However, not all objects in *SM64* can be clipped through as easily as others; walls, are hard to penetrate since their meshes are flat and sturdy. However, other objects, such as the game’s doors, are much easier to clip through.



Figure 10. *SM64*’s hub area viewed as a wireframe. Collision meshes exist on the triangles that make up the game’s digital geometry.

Doors are one of the main targets for skips in speedruns due to their abnormal collisions. The only thing needed to pass through them and achieve a skip is an immense amount of speed.

The type of speed matters; Mario's speed is connected to a value that can be positive or negative. Positive speed is gained when the player pushes the analog stick to steer Mario in a direction. Whenever the player moves Mario normally, he is considered to be moving "forward," and thus, positive speed can be considered as the value which increases when Mario moves in this way. Negative speed builds up when Mario moves backward, which is normally very hard to pull off, since the majority of cases where Mario moves by the player's input are considered "forward." However, compared to positive speed, negative speed has much more potential usefulness. Positive speed has a defined limit. Mario can only move so fast before that limit is reached and the game places a cap on his speed. Negative speed, on the other hand, does not have a set limit. If the player could manage to move backward without anything stopping them, they could far exceed the positive speed limit with negative speed (Ukikipedia, 2019).

This is where the BLJ, a glitch involving Mario's long jump move, comes into play. Normally, this move launches the player forward in a horizontal leap, useful for clearing gaps. Reversing a long jump by holding the opposite direction on the analog stick usually does little, aside from making the long jump considerably slower, shorter, and more awkward-looking. However, by finding certain objects in the game world - including stairs, slopes, and low ceilings - it becomes possible to repeatedly backwards long jump *into* certain kinds of collision meshes. This entails repeatedly inputting the long jump while holding backwards on the analog stick, causing Mario to appear still in the long jump pose. However, while the button is being mashed, Mario is actually gaining inordinate amounts of negative speed as long as he's locked in place. Once the player decides that they have gained enough speed by doing this, they release the stick, and Mario is sent sliding across the ground (and any gaps) at an almost blinding velocity.

Speed gained through BLJing makes it incredibly easy to launch oneself through doors, which can include big star doors. With the combination of gaining infinite speed and specific objects in the game having collision meshes which can be penetrated through use of that speed, it's no wonder that BLJing is widely considered such a useful tool in the speedrunner's arsenal. Speedrun uptake is arguably built around tricks like the BLJ, because those tricks remind that even without glitches or hacking, it is still possible to take up PRS arguments in all sorts of ways. Yet with runners' relationship to PRSes in mind, and their categories and techniques established, it must be wondered how those viewing *SM64* through speedrun uptake take up its arguments. How much is acknowledged and how much is reinterpreted, especially in the 0 Star category? What are the skips related to BLJs, how do they influence speedrun uptake of *SM64*'s PRSes, and what does adherence to this UL look like in action?

Speedrun Uptake and Full-Area Skips

With the rules of speedrunning now established, alongside the important glitches/skips, the foundation for speedrun uptake is now in place. Therefore, it becomes possible to analyze a specific speedrun and determine the ways that PRSes are perceived through speedrun uptake. This will be done by breaking down one of the fastest 0 Star original hardware runs of *SM64*, which is by a runner with the handle Xiah. Xiah finished *SM64* in six minutes forty-one seconds, an impressive feat considering that the average playtime for *SM64* is thirteen and a half hours (Howlongtobeat, 2019). Xiah's run is hosted on his YouTube channel Xiah7s alongside many other *SM64*-related videos, including many speedruns in other categories of rules and platforms. Being an accomplished speedrunner, Xiah's 0 Star run proves extremely useful for analysis, demonstrating how *SM64*'s PRSes can be taken up by those using the speedrun UL, along with how that uptake differs from the standard UL.

Generally, the comparisons between standard and speedrun uptake for this run can be framed in terms of the rules that either UL follows. Standard uptake must adhere to the rules argued by the game's PRSes. Speedrun uptake often varies in the PRS-argued rules it adheres to, but for the 0 Star N64 category in particular, the only PRSes that it appears to follow are those which are related specifically to the game's win-state - getting the final star in Bowser in the Sky. From there, the other rule that must be followed is use of an N64 console, which, unsurprisingly, many have experienced standard uptake while using. Differences between these ULs becomes apparent in video footage of either of them, and this is where Xiah's 2015 0 Star run video becomes incredibly useful. Following the video from moment to moment shows exactly what speedrun uptake looks like in action compared to standard (2015).

Unsurprisingly, Xiah's begins at a more frantic pace than a standard uptake playthrough. Where those demonstrating standard spend some time exploring or head to Bob-Omb Battlefield, Xiah does neither. Upon entering the castle, he ignores the argument of *SM64*'s locked doors - that in order to progress through the game, a player should be entering levels and seeking stars. Xiah goes to the staircase leading down to the keyhole door leading to area 2, or the basement. However, he cannot access area 2 yet. Despite the tools at his disposal, he still requires a key to get through this door, which is acquired in Dark World through the big star door back in area 1. How does he manage to get the key without a single star, ignoring the game's arguments for how he should play without any external tampering with the game?

The answer lies with the area 1 skip. Though not formally named, this skip clearly exists, for its use in the 0 Star category relies on specific steps and bypasses nearly all of area 1's levels, the exception being Dark World. Xiah fulfills the first step by descending the stairs to area 2, stopping halfway to the door, at which point he turns toward one of the two walls on either side.

Then, he jumps forward and grabs onto a seemingly non-existent ledge between wall and ceiling. He pulls himself up, clipping through the wall and ceiling, ending up inside the collision mesh of another staircase - the staircase in area 1 which leads up to the keyhole door leading into area 3. From this position, he initiates a BLJ aimed toward the entryway to area 1. With sufficient speed, he releases the BLJ, flying out of the staircase and rapidly sliding across the foyer floor.

Someone watching the speedrun could easily blink and miss the split-second he actually spends on the thoroughfare of area 1. His target is the area 1 entrance, and upon colliding with it, he clips into it (Figure 11). While the entrance to area 1 looks like any other door in the castle, it is normally a transition between inside and outside, which the game loads as separate levels.

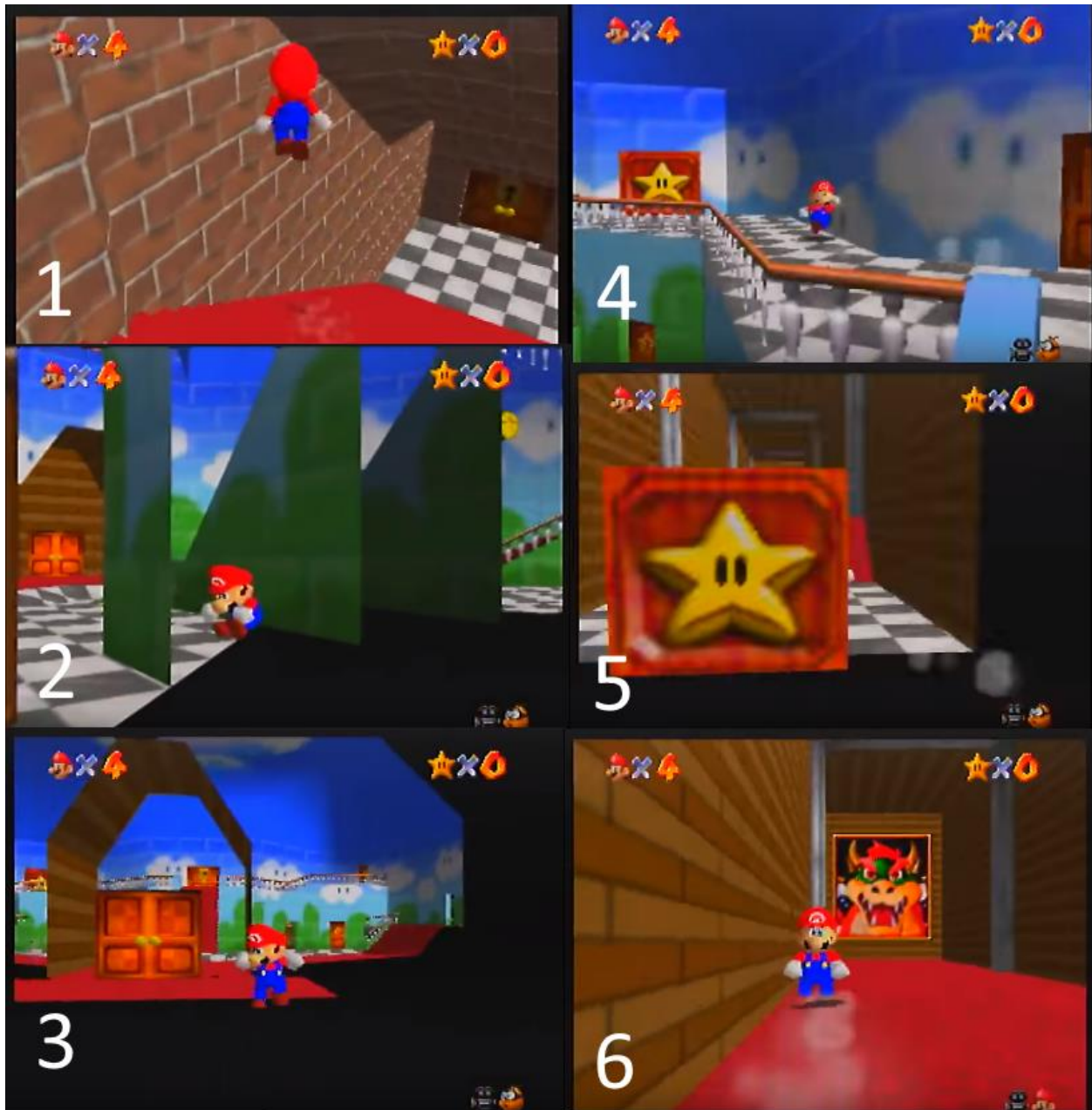


Figure 11. Xiah using a BLJ to perform the area 1 skip. The actual BLJ is displayed in 2. 3-5 display Xiah using the negative speed from the BLJ to clip through the area 1 entrance, before using that same speed to clip through the area 1 big star door. 6 displays the fruits of his efforts, which have led him to the entrance of Dark World without a single star.

However, because Xiah clipped through the door, he does not trigger the transition. Instead, he ends up in a void which exists “out of bounds” - separated from area 1’s collisions. These sorts of clipping-related glitches can frequently leave players stuck outside of levels, giving them seemingly no hope of returning aside from resetting their device or loading a save. However, runners make use of trips out of bounds for certain reasons, and Xiah is no different. He still has plenty of speed left from his BLJ, and by skilfully turning himself around in the void, he steers himself back into area 1 (also visible in Figure 11). Much like when he started his BLJ, he does not remain on the foyer floor for more than a couple seconds. He slides up the stairs to the big star door of area 1 and clips through it, which brings him to the entrance of Dark World. Worthy of note is the fact that Xiah’s run through Dark World does not make use of any glitches, and as such, he, like other runners, does not break any of the rules adhered to in standard uptake. He simply plays the level in the most efficient manner possible and swiftly defeats Bowser, acquiring the key to area 2.

In less than two minutes, Xiah manages to completely clear the entire first area of *SM64*, a task which would normally take significantly longer for players adhering to standard uptake. Xiah (and other runners) could be seen as ignoring the arguments of these PRSes because their uptake centers their attention on time commitment. Getting stars and opening doors is the way *SM64* is meant to be played, but with the skips runners have found, no one (perhaps not even the game itself) would argue the fastest method of reaching *SM64*’s win-state is playing it that way. Playing *SM64* the intended way all the way through takes thirteen and a half hours on average, after all. Even the 120 Star category is nowhere near that long. For Xiah and his speedrun uptake, completion rate trumps all. The game’s doors are “stepped over” like DeLeon’s tape maze.

The surface-level stability of *SM64*'s world is further revealed by the rest of Xiah's run. Upon reaching area 2, he immediately goes to perform the area 2 skip which, perhaps ironically, is actually considerably simpler (though by no means effortless) to pull off than the area 1 skip. Xiah simply heads to a conveniently placed railing near the area 2 big star door and does a BLJ, which allows him to speed toward the big star door and clip through it just like the one in area 1. Xiah also clips through the entrance to Dire, Dire Docks (referenced in Figures 2 and 3), which normally needs to be played through before reaching area 2's Bowser level, Fire Sea. Once again, Xiah skips everything in the area that can be skipped, only playing through Fire Sea because it nets him the key to area 3.

With the area 3 key in hand, Xiah quickly travels there. The area 3 skip is even simpler than the area 2 skip, since area 3 and area 4 are separated by a big star door and no Bowser level. The door is atop a flight of stairs, which lets Xiah BLJ his way up the steps and right into area 4. Area 4's big star door leads to the endless staircase, which in turn leads to Bowser in the Sky. The endless staircase is not as infinite as it seems, however - by BLJing up it, Xiah clips through the invisible trigger that would normally reset the player's position to the bottom of the stairs. This allows him to perform the area 4 skip with ease. Despite being at the very end of the game, this skip is arguably the easiest to pull off.¹¹ With area 4 skipped, Xiah is able to make it to Sky, defeat Bowser, and finally trigger the win-state - grabbing the final star and starting the credits. In speedrun terms, this is called "time," referring to the moment where a runner is allowed to stop their clock and record their completion rate. Xiah's time is six minutes forty-four seconds - not the world record, but only five seconds off.

¹¹ The area 4 skip, or endless staircase skip, was actually one of the first speedrun-related skips to be discovered in *SM64*. The first recorded instance of the BLJ was in the publication Club Nintendo in the year 2000. The other skips, which are difficult to perform, were discovered later.

Speedrun Uptake Motivation

Looking at Xiah's run, it is clear viewing the game through speedrun uptake means not being concerned with the arguments that the game is making about how it should be played. Runners have found a way of playing unique to them - one which they discovered through experimentation and applying their own rules in place of the argued ones. Speedrun uptake is the ability to take up PRS arguments not according to the rules they explicitly, internally prescribe, but rather, to rules generated by outside forces intent on playing the game as quickly as possible.

The simple conceit of external rules being applied to an existing game text has seemingly infinite potential for expansion. As Bogost (2016) often mentions throughout *Play Anything*, when applying one's own limitations to a system (his daughter avoiding cracks comes to mind), new states of play can be achieved. In a very lucid and perhaps overly ambitious interpretation, video games can be viewed as nothing more than complicated versions of cracks in a sidewalk. They simultaneously argue how they should be used (just walk) while also inviting alternative interpretations based on various motivations (don't touch the cracks).

These states of play are part of the reason that game rules should be viewed as rhetorical. Anyone can play a game according to the rules it argues they should be following, or they can take those rules up in whatever other way they want, even if that does not entail the same level of "game breaking" as in speedrunning. Self-imposed challenges like Nuzlocke in Pokemon games, where the player must (among other rules) release their captured creatures if they faint in battle, are proof of this. Beyond just speedrun uptake, many other uptakes for game rules certainly exist, all applying different limitations (or loosening them) for the sake of enjoyability.

This leads back to chapter 1 and the notion that no one experiences one form of uptake. Using a lens of speedrun uptake, standard uptake, or (in Pokemon specifically) Nuzlocke uptake,

is up to the player. They see the arguments that the game makes to them in terms of how they should be playing, and they choose whether to experience it as intended or to try something different. Again, the possibilities for how players choose to take up PRSes are virtually endless.

The possibility for many players to have many different kinds of uptake relates to the motivation behind speedrun uptake. Speedrun and standard uptake both draw a divide between uptake and practice - the latter being the player's ability to execute adherence to the rules they have chosen to follow. With standard uptake, this is rarely brought up as an issue since playing the game as intended is not exclusionary in its difficulty. Even played in their intended forms, some games, such as the *Dark Souls* series, have sparked controversy for being exclusionary in their difficulty, though *SM64* is generally not counted among them (2011).

The difference between uptake and practice in *SM64* becomes more significant when delving into speedrun uptake, however. As the skips performed by runners like Xiah show, speedrunning is not easy. Yet for runners, this creates one of the core motivations for the entire practice - the challenge of "pushing a game to its limits," as SpeedRunsLive puts it (2019). Speedrunning has been described as taking any game and turning it into a puzzle to see how to complete it in the fastest way possible, with all of a runner's uptakes being in the interest of finding the most optimal run (Lee, 2015). To play games like this requires a fair amount of skill, but to *view* games like this doesn't necessarily ask for a high level of dexterity with a controller. Case in point are audiences of speedruns, who supply another major motivation for the practice - the recognition that comes with having a world record.

Speedruns tend to draw in an enormous audience of viewers. Multi-annual charity livestream Games Done Quick is the best-known speedrunning event by far, seeing speedrunners from all over meeting up in one place to run their favorite games for charity. The January 2019

GDQ event (Congdon) amassed over 222,000 concurrent viewers at its peak and raised over \$2,000,000 for the Prevent Cancer Foundation and Doctors Without Borders. Speedrun viewers are far from a large demographic, but they are clearly passionate about runs. However, this is in spite of most audience members not actually being speedrunners themselves. They have seemingly acquired their knowledge of how speedruns work just by watching them, because without at least a rudimentary comprehension of speedrunning tricks and strategies, speedruns would look like nothing more than a confusing series of glitches to non-speedrunners.

Yet the viewers responsible for this added layer of motivation likely wouldn't watch speedruns if they had no idea what was happening in them. Speedrun uptake divorced from practice is frequently demonstrated by viewers of runs of games like *SM64* who are able to point out the tricks being used and how they are conducive to playing the game quickly. Even yours truly is far from a skilled *SM64* speedrunner, yet thanks to the wealth of runs recorded and documentation on glitches and skips available, I can easily follow what's happening in a run and enjoy it for the spectacle of seeing one of my favorite games deconstructed so thoroughly. Plenty of evidence (Figure 12) exists to show that non-speedrunners are still capable of understanding what is happening in a run even if they don't know the lingo (Harm Prins, 2017). When the idea of playing a game beyond its limits enters the mind of a speedrun watcher, they acquire a form of speedrun uptake. This UL may be underdeveloped and a little foggy, but knowing there are possibilities beyond standard is necessary for these runs to be enjoyed in the first place.



Harm Prins 2 years ago (edited)

My personal favorite moments (from a non-speedrunners perspective):

- 01:18. Glitching behind Chain Chomp's gate
- 02:07. Extreme parkour to the Caged Island
- 07:18. Getting the star in the sunken ship
- 13:39. Shortcut on the Princess's Secret Slide
- 18:30. Opening the cannon while racing Koopa the Quick
- 23:06. Picking up all the coins and the star in one shot on one of the hardest flying challenges
- 28:00. Glitching straight to the Cavern
- 31:41. Glitching into the pyramid
- 38:45. Extremely fast completion of "Bully the Bullies"
- 43:55. Shortcut to the wall-kick star in Cool Cool Mountain
- 45:16. Glitch and shortcut on the Cool Cool Mountain slide + bringing the penguin down really quickly
- 1:00:20. Lots of parkour to get the red coins in the bowser level
- 1:04:57. Who uses stairs anyways?
- 1:16:47. Two Shortcuts in Tall Tall Mountain
- 1:28:05. 100 coins in Tick Tock Clock
- 1:40:28. The final bowser fight, skipping all the shrinking of the arena

Figure 12. A comment on a *SM64* speedrun showing speedrun uptake by non-speedrunner.

Though not privy to the exact techniques being used, they have picked up enough speedrun uptake to see it is possible for *SM64* to be played using non-standard uptakes.

The desire to “push a game to its limits” has both a personal and performative outcome, particularly concerning audiences for speedrun content who, while possibly not skilled enough to execute speedrunning tricks themselves, are nonetheless capable of identifying the strategies used by speedrunners and enjoy it as a spectator sport of sorts. The uptake demonstrated by speedrun viewers is therefore somewhat comparable to that which is shown off by sports fans. Even if actually participating in the sport is impossible for the viewer, it is possible for them to understand how the sport functions to a degree that they can gain enjoyment from watching what is happening. Speedrun uptake in speedrun viewers is much the same; understanding how speedrunners view games by stepping back from standard uptake is fairly easy after one has either read a bit on speedruns or watched enough that it becomes apparent how they work.

CHAPTER IV: MODDER UPTAKE

Chapter Introduction - Modder Uptake

This uptake lens is named for the broadly defined practice of game modification, also known as modding. Modding entails altering the code of a video game to change part of it. “Change” could mean something as simple as altering the graphics or sound of the game without influencing the mechanics, environmental design, or other gameplay-related features. However, modding can also easily involve significantly more ambitious alterations to a game’s design, including changes to the gameplay which can have some interesting effects on a game’s rules. This leaves modders¹² in a very interesting position as far as uptake and procedural rhetoric go, since they negotiate the kinds of rhetoric that people who play their creations will encounter.

This is where the relationship between modder uptake and PRSes becomes apparent. Modders are known for creating what are called “mods” - software packages that function similarly to games and in some cases can even be sold as entirely separate pieces of software, like with *Counter-Strike*, which started as a *Half-Life* multiplayer mod only to spawn a franchise. *Counter-Strike* also has the unique distinction of having started with the base of a normally single-player experience in *Half-Life* and tweaking it to suit competitive multiplayer gameplay. However, mods like *Counter-Strike* are usually less common than mods that are smaller in scale, do not become proper franchises, and do not radically alter the gameplay of their game of basis¹³. One example of this is *MINERVA*, a mod of *Half-Life 2*. Like its aforementioned predecessor, *Half-Life 2* is a single-player first-person shooter. *MINERVA* plays about the same as *Half-Life 2*, albeit with most of its story conveyed through text (thanks to a smaller, or nonexistent, budget) versus its professionally-developed counterpart and its fully voiced characters.

¹² People who create mods.

¹³ A term used here to denote the software that a modification is derived from.

Aside from that distinction, *MINERVA* and *Half-Life 2* are hard to tell apart at a glance. *MINERVA* is based directly on *Half-Life 2*'s code, making use of the same environmental assets, enemies, weapons, items, and more (Figure 13). The main difference is in the levels themselves, which, despite using the same building blocks, are completely original in terms of actual design. Thus, *MINERVA*'s levels could be easily slotted into *Half-Life 2* without looking out of place. Mods like *MINERVA* are interesting for this reason, being customized software packages built on the code of existing games, offering new content for those who want more from those games, almost like a form of digital fan fiction. This comparison is quite apt for a mod like *MINERVA*, which tailors its level design and story to expand the *Half-Life* universe.



Figure 13. *Half-Life 2* vs *MINERVA*. Without detailed knowledge of either, telling them apart is difficult, as they share the same enemies, weapons, user interface, and environmental assets.

The intent behind *MINERVA*'s creation is well-documented by its creator, Adam Foster. In an interview, he stated his aim wasn't "to replace as much game content as possible; instead, it's to tell my own apocryphal story set in the *Half-Life 2* universe, and to actually release something for the public to play" (CVG, 2006). From Foster's explanation, it is clear that even beyond storytelling, the logic of fanwork that sticks to the conventions of existing work is here. Foster did

not want to create a new game using *Half-Life 2*'s code, a la *Counter-Strike*. Instead, he wanted to create a mod that felt at home within *Half-Life 2*'s world in story *and* gameplay.

From this, the basics of modder uptake are apparent. This lens views games and the logics that govern their worlds (and by extension PRSes) as being entirely reinterpretable, expandable, and transferable. A mod like *MINERVA* could be made for any game in existence, with a modder taking up PRSes and then considering how they could recreate those arguments, reinterpret them, or even reject them when designing a mod experience. Modder uptake blurs the line between persuader and persuaded for this reason.

Of course, it is noteworthy that all mod-related games mentioned so far are PC games. PC games are typically more mod friendly, being on a platform that is usually open to tinkering. Console games like *SM64* on the N64 are often harder to modify, since consoles usually are not so open to tinkering, and console games themselves have security measures that resist modding. However, this does not mean it is impossible for console players to demonstrate modder uptake. Even something like a child wondering about what a game would be like if X was replaced by Y, as simple as it may initially seem, could be interpreted as the early development of this UL. Beyond that, the era of emulators allows for games once only played on console to enjoy a new, extended life on personal computers, not only because they're widely available on the internet, but because emulators allow for previously console-exclusive games to be modified. Altogether, this creates a scenario where modder uptake can, in the modern gaming era, be felt by anyone, since more players than ever bear the knowledge that any game could be modded.

Continuing the fanfiction comparison from earlier, modder uptake could be compared to imaginative interpretations of characters and scenarios one might think of when reading a book. Similar to speedrun uptake, then, one is not required to be a modder to possess modder uptake,

just as one does not have to be a writer to think of ‘what if?’ scenarios for their favorite stories. For brevity, however, this chapter explores modder uptake exclusively from the creator’s side. While modders certainly take inspiration from other mods, the complexity of uptakes between different modders is too complicated to get into here. It is also for this reason that this chapter’s examination of the writers of digital fanfiction primarily concerns one work based on *SM64*, known as *Super Mario Star Road (SMSR)*, and its creator, who goes by the handle of Skelux.

This chapter locates evidence of Skelux’s modder uptake in the design of *SMSR* itself, which is clearly built to be an expansion to *SM64*, similar to *MINERVA*’s relation to *Half-Life 2*. This results in *SMSR* sharing many of the same rules, and by extension PRSes, with *SM64*, Skelux having recreated many of the same arguments with *SMSR*’s world design and mechanics. Additional evidence for Skelux’s modder uptake is provided by way of websites, forum posts, and YouTube videos by him.

With the writers of digital fanfiction out of the way, what of the readers and their uptake? Players, of course, have the ability to read and internalize a standard uptake lens for any game. Were they to play a mod of that game, they could make use of that same standard uptake lens, and their understanding of the rules would either be reinforced or subverted based on its design. This certainly can occur with *SMSR*, which, thanks to its similar design to the original *SM64*, almost seems like it expects players to apply their standard uptake lens gained from *SM64* to it. The result is similar for speedrun uptake, with similar skips functioning in *SMSR* and *SM64*.

However, it should be noted that unlike when discussing standard and speedrun uptake for *SM64* itself, discussing standard and speedrun uptake for *SM64* mods opens many doors that make the conversation more ambiguous. Even with *SMSR*, which Skelux has explicitly stated as being designed to evoke *SM64*, there are enough subtle differences between *it* and *SM64* that players of

the mod may develop standard and/or speedrun uptake lenses specifically for it. Beyond that, the simple fact that *SMSR* is a distinct entity from *SM64* makes it impossible to determine Skelux's intentions for sure, which means there is no way of telling whether the mod, as the previous paragraph suggests, is designed to where it "expects" players to make use of a standard and/or speedrun standard uptake lens originally developed for playing *SM64*. Even so, this does not prevent players from clearly using their *SM64* uptake lenses while playing *SMSR*. While moving beyond a game's original creators makes it hard to gauge the motives of players, creators, and everyone else surrounding a software's procedural rhetorical landscape, this chapter finds it useful to explore player intention from the perspective of using uptake lenses developed for the game of basis in a mod, but acknowledges that this does not paint a complete picture.

For this reason, standard and speedrun uptake lenses developed for *SM64* are viewed in action as evidence for the activities of digital fanfiction readers within this specific context. Standard and speedrun uptake relating to *SM64* can be applied when talking about *SMSR* since *SMSR* mostly recreates *SM64*'s rules, even if it technically subverts them in a few subtle ways. Like in chapters 2 and 3, evidence for standard and speedrun uptake is found in video content - in this case, let's play and speedrun videos for *SMSR*. This hints toward another possibility of lenses being derived from certain games - and that the PRS interpretations afforded by, say, *SM64* standard uptake can, in some cases, also be applied to modifications of *SM64* as well.

ROM Hacks - Mod Type Pertaining to *SM64*

Before getting into *SMSR* itself, it's important to consider the type of modification that is used for *SM64* and games like it. Obviously, making modifications for *SM64* requires emulation, which is the use of one computing device to recreate the operations of another. This can be handy

for replaying games for old consoles per their exact process environments, since the machines of yesterday are increasingly difficult to find in good condition, and repairing them can be costly.

The N64 has long been an extremely popular choice among emulation enthusiasts. However, recreating the operating environment of a console is one thing; getting games to run in that environment is another matter. Older consoles typically used proprietary physical media, such as cartridges, to contain their software information. The N64 did this, and as a direct result, the data for a game like *SM64* is maintained in an esoteric physical format inside a cartridge, written to be read only by N64. Thus, to re-play *SM64* on an emulator, the information inside the cartridge would need to be copied and rewritten in a form that can be stored on computers while also being readable by the emulator.

This is where ROM¹⁴ images - or ROMs - come into play. ROM images are digital copies of the physical data in game cartridges. When loaded on an emulator, a ROM runs in a way that closely approximates using the original console. Of course, while the ROM usually contains the unaltered code of an old game, one could feasibly change the code of a ROM before running it. The emulator will run the modified ROM as if it's the original game, but the actual play experience will be modified. This is what is known as ROM hacking.

ROM hacking is defined as “Modifying the data in a ROM image to achieve such purposes as playing the game in a different language than intended, creating new levels for old games, or maybe playing with a different skill level than intended” (Romhacking.net, 2019). Modifying both actual levels and skill levels appears to be the most eye-catching element here - taken to an extreme, it would seem that a modder could create a new version of an old game.

¹⁴ ROM stands for “read-only memory.” ROM emulation involves copying this “read-only memory” from the chip or other physical media it is stored on into data.

Super Mario games have enjoyed a long history of ROM hacking for everything from *Super Mario Bros* on NES emulators to, naturally, *SM64*. *SM64* ROM hacking has been going on so long that tools specifically for modifying the game have been created, such as Toad's Tool 64 (Figure 14). This allows users to create simple edits to *SM64*'s code for various effects, such as adding enemies or hazards where none existed before. These edits do not introduce significant changes to the asset pool¹⁵, instead making use of the assets already in the game.

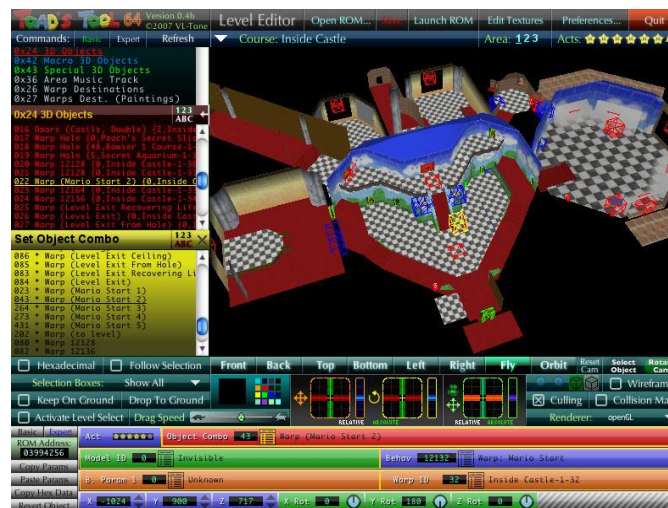


Figure 14. Toad's Tool 64, a modification tool for *SM64*.

After users of Toad's Tool 64 are satisfied, they can save them as a custom ROM image, which is what is known as a "ROM hack" (or just "hack"). *SM64* hacks vary in complexity, with the simpler ones doing little more than editing enemy and object placements, often for the sake of (as the Romhacking.net definition implied) increasing the game's difficulty. These hacks are simple to explain visually since they are set in the same environment as the original game but with tweaks. One example of this can be seen in Figure 15 with *Kaizo Super Mario 64*, a ROM hack which increases the density of enemies and hazards in the game. These hacks can be fun for those seeking a difficult challenge, but they do not change anything fundamental about *SM64*'s design.

¹⁵ Enemies, objects, textures, and other "building blocks" of a game that the player interacts with.



Figure 15. *Kaizo Mario 64* is a ROM hack that increases *SM64*'s difficulty. This is partly done by adding hazards, such as filling the normally safe Area 1 with fire.

More ambitious ROM hacks are those that *do* make large changes to *SM64*'s world. These are ROM hacks that do not merely rearrange the objects in *SM64*'s existing play spaces, but throw those spaces out in order to build new ones. Of course, while the designs of levels in these sorts of hacks are unlike anything in the original, the core programming does not change, and the asset pool remains largely (often entirely) the same. These ROM hacks operate off of the same collision detection, physics, and other mechanical features as *SM64*, while also containing the same objects, enemies, and other elements. They could be compared to demolishing an existing building, then using the material to construct something new. The architecture is different, but the building blocks beneath it are the same as the predecessor. This can be seen in Figure 16, which displays a level from *SMSR*, compared to Figure 17 and 18, which display the original game.



Figure 16. A screenshot of a level from *Super Mario Star Road*. The mod contains new levels that are almost entirely comprised of re-used assets from *SM64*.



Figure 17. Compared to a screenshot of Bob-Omb Battlefield from *SM64*. Notice the tandem platforms, trees, and grass.



Figure 18. Compared to a screenshot of Princess Peach's Castle from *SM64*. Notice the trees, grass, and the brick texture on the castle.

Considering ROM hacks like *SMSR* are built off of the original *SM64*'s engine and assets, it is only natural to assume they might port over some of the original's rules and, by extension, PRSes as well. What kind of procedural arguments are made by a ROM hack that takes the assets of an existing game and uses them to create a new experience? This question is one that has many answers depending on the design of the ROM hack and the intentions of its creator. Luckily, when discussing *SM64* ROM hacks, there are many examples of hacks that are designed with the specific intention of recreating *SM64*'s rules.

A great hack to explore this idea is the one that has been discussed up until now - *SMSR*. This hack was created with the intent of generating new content for the classic *SM64* experience, serving as an unofficial sequel of sorts (Skelux, 2011). *SMSR* manages to recreate the rules of *SM64* with amazing accuracy despite its actual levels being completely new. This makes it ideal for an analysis relating to rules and PRSes, since it almost serves as a modded mirror to *SM64*.

Rules, Progression, and PRSes in *Super Mario Star Road*

Similarities between rules and progression in *SMSR* and *SM64* start with their hub areas. *SM64* has Peach's Castle, a hub divided into four areas based on their elevation in the structure, frequently referred to as the first floor, basement, second floor, and third floor for that reason. *SMSR* has Star Leap Tower, a hub also divided into four areas distinguished by their elevation, though the Star Leap's equivalent to *SM64*'s first floor is outside the structure rather than inside. This is one of the many small changes that *SMSR* makes to the hub structure of its game of basis, though overall, *SMSR*'s hub is nearly identical, leading to very similar progression between the original and the ROM hack.

SMSR is based around the same general progression system as *SM64* - players are meant to enter levels, get stars, and once their star count rises high enough, they advance to new areas.

However, *SMSR* vastly increases the number of stars needed for access between different areas. To get from *SMSR*'s area 1 to *SMSR*'s area 2, for example, a player must raise their star count to twenty - twelve more than the original game's eight. This does not significantly change the rules, but it could argue to the player that they should thoroughly explore each level to a greater degree than is argued in the original. *SM64* minimally requires players to visit only one proper level to advance to area 2 thanks to a secret star¹⁶ in area 1, whereas *SMSR* requires the player to visit at least three proper levels before making progress. Additionally, unlike *SM64*, *SMSR* locks two of its area 1 doors behind an eight star requirement, further incentivizing an extremely close and intimate exploration of an area's levels.

After area 1, most levels in *SMSR*'s area 2, 3, and 4 become available automatically, though this is also fairly similar to *SM64*. All these similarities are partially due to the ROM hack needing to directly replace each level in the original game with its new levels when it overrides the original's code. As a result, the two titles actually share the same number of levels. While *SMSR* rearranges the exact number of levels per area, the actual count is the same, as can be seen through comparison of Figure 19 in this chapter to Figure 3 from back in chapter 2. The identical level counts further demonstrate similar rules and progression between both titles. However, this is both further solidified and slightly challenged when taking *SM64* and *SMSR*'s Bowser levels into consideration.

¹⁶ *SM64* contains a handful of small secret levels that only contain one or two stars, compared to the seven available in the full-sized levels. *SMSR* has equivalents to these as well.

**Each level contains seven stars,
used to unlock more levels and areas**

| | |
|---|---|
| Area 1 Access requires - 0 Stars | Area 3 Access requires - 40 Stars |
| Bob-omb Islands Unlocked with - 0 Stars | Colossal Candy Clutter Unlocked with - 0 Stars |
| Sky Land Resort Unlocked with - 0 Stars | Cloudbail Station Unlocked with - 0 Stars |
| Piranha Plant Pond Unlocked with - 0 Stars | Fatal Flame Falls Unlocked with - 0 Stars |
| Chuckya Harbor Unlocked with - 8 Stars | Area 4 Access requires - 65 Stars |
| Gloomy Garden Unlocked with - 8 Stars | Bob-omb Battle Factory Unlocked with - 0 Stars |
| Area 2 Access requires - 20 Stars | Starlight Runway Unlocked with - 0 Stars |
| Colorful Coral Caverns Unlocked with - 0 Stars | End of game Access requires - 80 Stars |
| Large Leaf Forest Unlocked with - 0 Stars | |
| Melting Snow Peaks Unlocked with - 0 Stars | |
| Mad Musical Mess Unlocked with - 30 Stars | |

Figure 19. SMSR progression diagram (text) - compare to Figure 3 from CH 2.

Like *SM64*, *SMSR* has three Bowser levels - one for area 1, one for area 2, and finally, one for area 4, which ends the game upon being completed. Both titles are missing a Bowser level between area 3 and 4, likely due to the aforementioned limitation that *SMSR* adheres to. Bowser levels are still locked behind star doors, and upon being completed, they offer keys that the player can use to open up more of the hub. *SM64* has Bowser in the Dark World for area 1, Bowser in the Fire Sea for area 2, and Bowser in the Sky for area 4. *SMSR* has Bowser's Slippery Swamp for area 1, Bowser's Retro Remix Castle for area 2, and Bowser's Rainbow Rumble for area 4. These levels correspond to each other fairly closely for the most part, though *SMSR* defies the rules and connected PRSes of *SM64* by having the only fight against Bowser in Rainbow. The two preceding levels, Castle and Swamp, have newly designed encounters involving existing enemies slotted in. This arguably increases the variety of what Bowser levels offer in the hack, which might argue to the player they should expect something different in each boss fight.

With the rules and progression in *SMSR* being nearly identical to ones within the game the ROM hack is based on, the PRSes in *SMSR* are, unsurprisingly, also very similar. The PRS of the door is in *SMSR*, though its textual presentation is a little more vague compared to *SM64*. Approaching the key door leading to area 2 without actually getting the key results in the player being rejected followed by a message that ambiguously states "You cannot open this door yet." Of course, as was mentioned in chapter 2, even without this text the door would still cause most players to infer that they should be entering levels and acquiring stars until they can acquire the key that allows them to unlock the next area. This is effectively the same argument as in *SM64*, though it could be argued it is somewhat different since *SMSR* has a handful of stars in its hub.

The number of stars accessible in the hub is fairly low overall, however, so players will inevitably still have to visit the proper levels in order to make true progress, and the argument of

the door PRS remains the same. Nonetheless, any minor change to *SM64*'s procedural arguments in *SMSR* should be noted. Some of the more noticeable changes can be found in *SMSR*'s versions of *SM64*'s boss fights. While *SMSR*'s boss enemies are the same as the ones from *SM64*, aspects of their physics and artificial intelligence (AI) differ. For example, King Bob-omb - the boss fought in Bob-omb Battlefield in *SM64* to get the first star - returns in *SMSR*'s BoB replacement, Bob-omb Islands, with a few upgrades.

In both *SM64* and *SMSR*, the player defeats the King by lifting him from behind and tossing him, the impact from his heavy landing damaging him. However, *SMSR* makes a slight change to his animations and physics. In *SM64*, King Bob-omb sailed gingerly through the air, likely to give satisfying feedback to his landing, which is accentuated by weighty sound effects. He also spends about a second on the ground recovering, likely to further sell the effect.

In *SMSR*, King Bob-omb spends far less time airborne, dropping like a rock when tossed. He also recovers from being thrown nearly instantly. This increases the speed of the fight, allowing skilled players to defeat the boss faster if they know what they are doing since they won't spend nearly as much time watching him flying through the air and slowly getting back up. King Bob-omb also takes an extra hit in *SMSR* to defeat - four as opposed to *SM64*'s three.

These changes are applied to all of the game's bosses. Across the board, they sport faster animations and additional hits to take down, and some even have their attack patterns changed to make them more intense. The intention seems to be making bosses both faster to defeat and more of a challenge to the player, creating a new experience fighting old foes.

Standard and Speedrunner Uptake of *SMSR* PRSes Based on *SM64*

Within this section, several gameplay videos, some commentated, some uncommentated, are used to show the ways players take up the PRSes in *SMSR*. The first few examples of PRS uptake focus on players applying the standard uptake lens that they learned in *SM64* to *SMSR*, which works for the most part, but is occasionally challenged by certain design elements within *SMSR* that are changed, such as the boss characters. Players are left to take up PRSes in an environment that is simultaneously welcoming and familiar while also frequently proving alien and subversive, resulting in a play experience that is unpredictable even for *SM64* veterans.

The more basic arguments comprehended via standard uptake certainly exist in *SMSR*, with many “blind” let’s plays of the ROM hack demonstrating this. One of the more interesting examples is a playthrough by SullyPwns, which makes use of a multiplayer modification for the ROM hack (yes, mods of mods are a thing). This modification does nothing more than allow multiple players to play *SMSR* at once. An identical mod exists for *SM64* too, though naturally, it’s easiest and most convenient to install and use it on an emulator.

SullyPwns’s multiplayer playthrough of *SMSR* is intriguing because while Sully himself is new to the hack, his friend and multiplayer partner Aaron is somewhat experienced with it. This leads to a dynamic that is similar to what is seen in the Fine Bros video cited in chapter 2, with an experienced and inexperienced player demonstrating a shared understanding of standard uptake as the way the ROM hack is meant to be played, a premise which is itself owed to the way that *SM64* is meant to be played. This is shown in a similar way to Flapper’s video, also cited in chapter 2, as Sully and Aaron approach the key door leading into Star Leap Tower.

Sully instantly draws a connection between the tower itself and *SM64*’s equivalent to it, Peach’s Castle, recognizing the function of both as the game’s hub. This quickly confirms that

while Sully has not played *SMSR*, he has played *SM64*, and he understands the rules that surround the progression system of the hub area in both the original game and the ROM hack. Additionally, he is trying to apply his understanding of *SM64*'s PRSes by way of standard uptake to *SMSR*, which proves a completely valid interpretation of the game's rules when he approaches the key door and makes contact with it, triggering the message "You cannot open this door yet."

Upon receiving this message, Sully tries to say something, but Aaron talks over him with a seemingly reflexive statement: "Yeah, so we can't get into there until we beat the first mini-boss." By "mini-boss," he refers to the "boss" of Swamp, which is a room filled with the piranha plant enemy from *SM64*. Sully understands Aaron's meaning instantly, and both players then decide that they should "go to the first world," in Aaron's words. This choice of words is interesting since, unlike in *SM64* where only Bob-omb Battlefield can be accessed in at the start of the game, *SMSR* allows players to access a multitude of area 1's levels at the beginning, meaning that there technically is no "first world." Despite this, the rules argued by *SM64* - which include the rule that at the start of the game only Bob-omb Battlefield can be explored - seem to have been thoroughly imprinted in both players' minds. This holds true in regards to the door PRS as well. Whether both Sully and Aaron knew the rules they'd be following in *SMSR* before starting is impossible to determine, even if it is fairly evident that they have both played *SM64*. Irrespective of that, it is clear that when a locked door showed up in the hack, both of them instantly understood what the ROM hack was arguing they should do.

Of course, players' expectations of rules in *SMSR* based on their experiences with *SM64* do not prepare them for everything the ROM hack has to offer. Upon reaching King Bob-omb, the boss of Bob-omb Battlefield in *SM64* and Bob-omb Islands in *SMSR*, Aaron points out to Sully that the boss now requires four hits to be defeated as opposed to three in the original game. Both

players acknowledge this as a minor subversion of the rules in the original game, though they do not verbally acknowledge the animation speed increase.

This may be down to both players predominantly viewing the hack through a standard uptake lens gained when playing *SM64* while also playing it in that way. While Sully and Aaron might have noticed the animation speed increase, they might not have felt a need to mention it either because they were uncertain about whether it truly was faster or because such an element of gameplay does not fall into the concerns of standard uptake. Players viewing the game through a speedrun uptake might have different things to say, however. In his review of the hack, speedrunner Nathaniel Bandy points out every change that was made to King Bob-omb, including the animation speed increase. He also describes these changes as universally being “improvements” to the fight, though whether or not he feels this was simply because they make the fight go by faster is up to speculation or direct questioning (Nathaniel Bandy, 2018).

Other speedrunners have interesting uptakes of the game’s rules as well, many of which hint at how rules can potentially be disrupted by hacks and the uncertainty that results from this. Speedrunner DGR, in his blind playthrough of *SMSR*, reached a point in Bob-omb Islands where he was unsure of how to progress. Spotting a sloped surface that could work for initiating a BLJ, he questions whether he is expected to use a BLJ by the game’s developers - something that no one would ever consider when playing *SM64* (DGR 2018).

Complicating matters is the existence of speedruns for hacks, *SMSR* included. *SMSR* runs and information surrounding them show that speedrun uptake developed playing *SM64* can be applied to *SMSR*. Similar categories for *SMSR* runs exist, such as 130 star, 80 star, and the now familiar 0 star. 0 star in particular plays out in almost an exactly identical way; though the skips

and glitches used to achieve them are sometimes different from *SM64* in *SMSR*, the result yielded by these skips is the same.

Skipping areas in *SMSR* is in some cases easier than in the original. In SimpleFlips's 0 Star speedrun of *SMSR* (2015), it's shown that where the area 1 skip in *SM64* requires a complex series of clips and a BLJ, the one in *SMSR* only requires a simple clip through the door leading into Swamp. The area 2 skip is more equal in difficulty between *SM64* and *SMSR*, as both require a BLJ to clip through a door. The area 3 skip is more complex than it was in *SM64*, since *SMSR* requires the player to unlock a cannon and fire themselves out of it to reach area 4. However, with a few well-placed BLJs, it's possible to get inside the cannon and fire it without properly unlocking it. *SMSR* lacks *SM64*'s endless stairs before its climactic Bowser level, but both the game and the hack only require a BLJ to be done to bypass the barrier to their ending.

The same idea of ignoring certain rules and seeking out means of moving as quickly as possible applies in both *SMSR* and *SM64* in seemingly the same way. Speedrun uptake simply entails making use of whatever glitches and exploits are available to beat a game faster, and as such, unlike the playthrough by Sully and Aaron, players like SimpleFlips enable themselves to break rules in order to reach a faster completion time. There is a mixture of mostly old but with sprinklings of new when it comes to PRS uptake of these softwares' rules.

Modder Uptake

Of course, it must be asked how modder uptake led to this mix of old and new in *SMSR*. What unique features of this lens made Skelux able to design this hack in a way almost perfectly recreating the rules and progression structure of *SM64*, right down to specific PRSes like doors, while also transforming a handful of those rules and PRSes for what appear to be specific ends? Skelux's exact reasoning may never be truly known, but it can be speculated from some of his

statements, forum posts by him and others, and the design of the hack itself, that his modder uptake had some noteworthy influences. Specifically, those influences seem to arise in the form of Skelux’s understanding of standard and speedrun uptake in regards to *SM64*.

Standard uptake is fairly easy to trace via evidence from the hack itself alongside the descriptions of certain YouTube videos where Skelux explicitly states his intention to recreate the “classic *SM64* level style” in *SMSR* (Skelux, 2011). It is clear from players applying their experience of standard uptake in *SM64* to *SMSR* that the hack welcomes this type of uptake. Skelux seemingly intended this interpretation of the hack’s rules, and based on existing evidence, it seems he deliberately recreated PRSes like that of the door for that reason. That he uses his own unique door assets (Figure 20) rather than reusing the ones from *SM64* but still chooses to structure their procedural argument in the exact same way is testament to this. In order to have done this, Skelux’s modder uptake would’ve needed to connect his understanding of standard uptake of *SM64*’s PRSes (likely from playing the game many times himself) to a larger meta understanding of game rules as arguments detached from games themselves. He needed to know how to recreate these rules to argue to others that which was argued to him.



Figure 20. A big star door in *SMSR* (left) vs *SM64* (right). While different in visual design, they are functionally identical.

Of course, this is not to say that modder uptake, or even simply Skelux's modder uptake, is limited to just wanting to recreate *SM64*'s rules. Skelux obviously took some liberties, however slight they may have been, with the rules. There are several interesting examples of this, with one in particular being discussed in the final section of this chapter. For now, the incorporation of what appears to be speedrun uptake elements filtered through the rule-morphing lens of modder uptake is focused on.

There are many possible explanations for what led Skelux to have the idea to change almost nothing about *SM64*'s enemies in *SMSR* except for the bosses. Yet based on the positive responses from speedrunners, it seems that Skelux may have recognized an alternative uptake for the boss related PRSes. Maybe instead of looking for a sense of "impact" from the animations, Skelux took up the PRSes surrounding *SM64*'s boss fights in a way where he wondered why they weren't more efficient to provide a gameplay experience with the least amount of down time possible. Maybe the reason he took them up this way was because he was trying to throw speedrunners a bone, recognizing that those who play in the 80 and 130 star categories might want boss fights that are faster and more challenging than those in the original.

For those who play in the 0 star category, it's possible Skelux intentionally left in glitches that cropped up during development due to his knowledge they would be used by speedrunners. *SMSR* has been updated many times, yet these skip-enabling glitches remain untouched, despite them having been used by speedrunners for years. If Skelux left in means of subverting his own PRSes for the sake of speedrunners, his modder uptake would need to be capable of absorbing and redirecting/recreating the thoughts of speedrunners too. This further situates modder uptake as having a meta awareness of rules.

Modder Uptake Motivation and Motivation of Mod Players

Adam Foster, creator of *MINERVA*, has voiced his dissatisfaction with the design of *Half-Life 2*'s levels on the basis of them not being architecturally realistic. While *Half-Life 2*'s levels are well-made from a pure game design standpoint, Foster argued that because they are so fine-tuned for gameplay purposes, they “resemble an unconnected series of boxes” when divorced from the original context in which they were created (Watson, 2006). Foster hoped to improve this aspect of *Half-Life 2* by designing levels as places first and game levels second, and for the most part, he apparently succeeds in this aim.

Of course, as was stated at the start of this chapter, Foster's intentions did not include changing the base gameplay of *Half-Life 2*. Thus, while he might've made the world more realistic, the gameplay remains untouched, Foster seemingly agreeing with the arguments the game posed about how it should be played mechanically. Yet in challenging its level design, Foster was motivated to take up *Half-Life 2*'s argument in a way that he retooled it to his liking. This is one of the main motivations that can crop up around modder uptake - taking up one of the game's arguments to such an extreme that a new experience is created to fill a deficiency.

This motivation can be found in *SMSR* as well. Skelux's aforementioned desire to build *SMSR* to encapsulate the “classic *SM64* experience” is paired with his desire to mix in a more “open” or “sandbox” design ethos. He wants to make the game more open to player choice while maintaining the same general structure, and in this regard, he succeeds. As was mentioned before, *SMSR*'s area 1 immediately allows players to select from three levels when they first start the game, not to mention other secret levels and stars hidden in the hub. This obviously fulfills Skelux's desire to make the game more open, yet its overall progression does not deviate significantly from the HUB area traversal seen in *SM64*.

For players of mods, motivation to play can be somewhat similar. If they enjoy a game, they have clearly become accustomed to its rules and accept its arguments relating to those rules. However, players can only replay the game that “sold them” on those rules so many times over. They might desire more of the same, or a more challenging version of the game that they enjoy, or, in some cases, a version that addresses what they perceive as flaws from that same game. *SM64* fans have options like *SMSR* for this purpose, and in playing them for the above reasons, they will most likely bring the uptake lens(es) they developed for *SM64* in with them.

Of course, these ideas regarding the motivations of modders and the players of mods are, as was expressed in this chapter’s introduction, speculative due to the ambiguity surrounding intention when it comes to mods. Some players might go into *SMSR* without any preconceived notions about how it will play whatsoever, proceeding to develop a standard uptake lens for the mod rather than the game of basis. Some modders might not have any grand ambitions, but are simply modding to see what they can get away with. Moving into the conclusionary fifth chapter, additional possibilities for mods are explored, though only to further speculate their implications.

CHAPTER V: FUTURE IMPLICATIONS

Chapter Introduction - Conclusion and Future Implications

This implications section explores more possibilities related to uptake lenses and PRSes, including directions for future research. Therefore, this section is divided into three subheadings, which explore implications for standard uptake, modder uptake, and the self-imposed challenge, or “SIC,” an uptake lens that is hinted at in certain parts of this thesis but not properly elaborated. Any of these subheadings asks questions and introduces complications that could be the topic of their own research paper, and they are thus discussed here only as possibilities and not as foregone conclusions. Uptake lenses provide limitless potential for examining how people take up rhetoric from games, and it should be noted that even the implications that are discussed here do not represent the full range of possibilities for procedural rhetorical systems theory. For now, however, these three subheadings bring up some complications and questions that will likely need to be addressed by procedural rhetoric in the future.

The standard uptake subheading discusses how PRSes can appear in multiple games, often in the same game series. These PRSes tend to argue the similar rules within each game, falling under the label of “series PRSes.” Series PRSes are taken up by the standard uptake lens, and if a player plays enough games in the series, they could develop “series standard uptake,” which leads them to anticipate the same argument whenever they find a similar-looking PRS. This can be complicated when what appears to be that “similar-looking” PRS makes a different argument altogether, which can potentially subvert player expectations in a frustrating manner. From this, there appears to be an incentive for game makers to maintain series PRS consistency, knowing that their fans will expect certain arguments from them.

The modder uptake subheading discusses complications posed by mods that are not mentioned in chapter 4. The main issue that comes up is the link between PRSes and game assets (individual components of a game, such as enemies and obstacles, which are abstracted as code). Changing just a small aspect of a game’s code could radically alter the arguments it presents, which is very much visible in mods that try to re-create mechanics and rules from other games. *Super Mario Bros. 64*, for example, attempts to recreate levels and mechanics from the original *Super Mario Bros.* in *Super Mario 64*’s engine. Numerous elements in *Super Mario Bros. 64*, such as power-ups, are very obviously designed to evoke their *Super Mario Bros.* incarnations. PRSes attached to these power-ups are similar to those from the game that they are adapted from, but differences in programming make it so that the ones in *Super Mario Bros. 64* cannot convey the same argument as the ones in the “real” game. This leads to currently unanswered questions about the importance of game asset code in relation to the arguments it can represent.

The last subheading explores self-imposed challenges, or SICs, as an uptake lens. Self-imposed challenges see players taking up PRSes in such a way where they choose to apply more limitations than what those PRSes actually argue for. This could include ignoring helpful items despite their use clearly being intended by the game’s design. Any player can create a self-imposed challenge, and in doing so they are choosing to take up a game’s arguments in a way where they decide they do not provide a satisfying challenge. Applying additional rules to work around forces players to be creative, which can lead to interesting discoveries, such as one player who managed to use parallel universes in *Super Mario 64* to reach a star without jumping. This desire for challenge, which escapes and narrows arguments from others and the world itself, is something that drives not just gamers, but people in general to discovery.

Implications of Standard Uptake - Rules and PRSes for a Series

In 2013, YouTube channel Game Grumps posted a playthrough of *Super Mario Sunshine* (*SMS*), a game released in 2002 for Nintendo GameCube that serves as a follow-up to *SM64*. Arin Hanson, the game-playing online personality who usually plays games for Game Grumps, was the one to tackle *SMS* for the channel. Though he has stated many times he enjoys the series, he had not played *SMS* before the 2013 playthrough, making it an artifact of his initial thoughts. Throughout the playthrough, it is clear that Arin brought many expectations for what he assumed he'd find when playing a *Mario* game. Specifically, he thought certain gameplay elements would work the way they do in other *Mario* games, and while he was proven correct in many instances, there were also many times where his expectations were heavily subverted.

One example of this subversion is seen in part 42 of the Game Grumps *SMS* playthrough, where Arin encounters the ghost-like Boo enemy. In most *Mario* games, enemies the player faces can be defeated in various ways, with the most common means available being to jump on them. This creates an argument as to the offensive capability of the player's jump on most enemies; however, some enemies cannot be defeated in this way (or in most others), including the Boos. *SMS* breaks this tradition, however, finally letting players defeat this enemy by jumping on it. Arin, upon initially encountering Boos in *SMS*, does not realize that he can do this and tries to avoid them rather than jumping on them, which leads to him being harassed by a gang of ghosts. He eventually realizes he can jump on Boos, and his commentary during this realization is tinged with catharsis and slight annoyance. His own words, "Historically, you can't jump on Boos," directly explain why he is annoyed; his expectations were based on experiences not with *SMS*, but with games *other* than *SMS* (2013).

Arin's rule expectations did not originate from his uptake of *SMS*'s PRSes in a vacuum, and as such, he clearly was not making use of an uptake lens developed just from playing *SMS*. The lens he was using was definitely standard, as it was seeking how *SMS* is meant to be played, yet it appears to rely on knowledge developed while playing multiple other *Mario* games. Apparently, after experiencing the argument of "you cannot jump on Boos" in so many games, Arin developed a standard uptake lens for the entire *Mario* series that anticipated that argument. Having apparently never seen the Boo argument conveyed differently in any other *Mario* game, he anticipated it from *SMS*, only for the argument to be different from what he expected.

Arin's experience is far from unique. Anyone who has played a game series knows that, in everything from mechanics to visuals to music, continuity is maintained across multiple titles. Expecting PRSes to function the same in each game based on aspects like their visual design, which was certainly the case with Arin's experience upon seeing the familiar design of the Boo, seems to be connected to an uptake lens that goes beyond being applicable to a singular game. *Series standard uptake*, then, seems to be the lens that anticipates these series-wide arguments.

Having existed for multiple decades and containing numerous pieces of iconic imagery, the *Mario* series has many PRSes that can be viewed under the lens of series standard uptake. Obviously, not all examples of *series PRSes*, as they can be called, are subversive like in *SMS*. After all, Arin would only have assumed he could rely on his antecedent knowledge of the series if other rules and PRSes in *Mario* games had been presented in a more consistent fashion before. This level of consistency appears to be maintained by game designers for a variety of reasons, and in some cases, the presentation of PRSes that use familiar visual and mechanical arguments seems to be done specifically in anticipation of series standard uptake like Arin's.

One example of this is seen in two elements of the *Mario* series that are closely related - blocks and power-ups. The ways blocks and power-ups function are both incredibly well-known, to the point it could be said their mechanics are just as iconic as the character of Mario himself. Both are visible in Figure 21, which shows them in the game their relationship was established in - *Super Mario Bros (SMB)*.

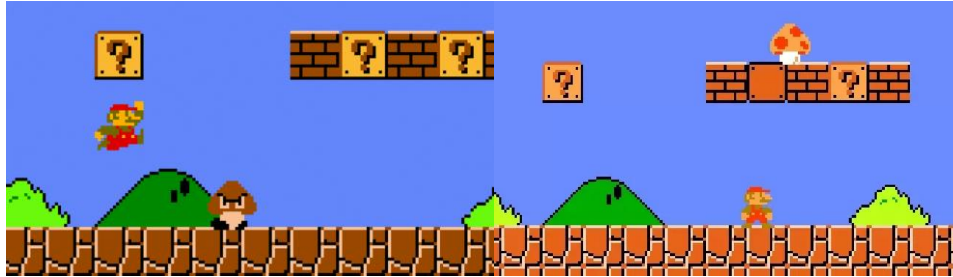


Figure 21. Mario jumping at a question block to get a power-up.

SMB contains most of its power-ups inside of the famous “?”-marked question blocks. The player strikes these blocks from below with a jump, which usually only grants them a coin, but in some cases, it grants a power-up. Question blocks argue that jumping is useful not just for defeating enemies and crossing pits, but for acquiring useful resources as well. These resources are linked to a system of randomness, which further argues to the player they should be constantly striking question blocks to find power-ups since they never know what they’ll get. The power-ups themselves also serve argumentative purpose, giving the player enhancements for use during gameplay. The aforementioned (and pictured) super mushroom is the most obvious, as it allows the player to sustain an extra hit that would normally have led to them losing a life. This further tightens the game’s argument that one should constantly hit blocks for power-ups, given that they are useful no matter what situation a player might find themselves in.

These arguments are repeated and expanded upon in several subsequent *Mario* games. *Super Mario World (SMW)* still retains question blocks and their associated randomness element,

alongside the usual argument that jumping at them from underneath can yield useful power-ups. The power-ups that were available in *SMB*, including the super mushroom, all return in *SMW*, alongside the new cape feather. This feather increases the diversity of the power-up pool while serving as a natural evolution of the argument of power-ups always being useful tools to have. The feather not only allows the player to sustain an extra hit, but slows the descent of jumps, making it useful for landing hits on enemies and ensuring one doesn't fall into bottomless pits. Once again, the game argues that one should be hitting blocks to get always-useful power-ups.

SMW also expands PRSes surrounding blocks and power-ups by adding colored blocks, such as yellow and green blocks. These blocks will always contain the same specific power-ups, with the yellow blocks containing super mushrooms and green blocks containing cape feathers. Designing colored blocks in this way removes the randomness argument seen in question blocks, which seemingly overrides that argument since colored blocks can be counted on for consistency. However, colored blocks avoid muddling the argument of question blocks by being much rarer, meaning the player is still encouraged to hit plenty of question blocks from below like in *SMB*. Additionally, colored blocks serve a totally new function: giving the player a reason to explore. Colored blocks are always first encountered as unfilled, intangible outlines, and in some cases, these outlines form a bridge leading to a location that is inaccessible by most other methods. Filling these outlines requires the player to find the corresponding switch, which not only gives them access to a consistent source of a certain power-up, but may (as in the case of Figure 22) provide them with a means of reaching new areas.



Figure 22. Filling unfilled block outlines in *SMW*.

With blocks and power-ups, the *Mario* series has developed a pair of staple series PRSes that seem posed to appear in every game. Even if they might have taken on some new elements in games like *SMW*, the basic arguments of jumping at blocks from below for the possibility of getting a power-up, and using that power-up to enhance normal gameplay, remain consistent. *Mario* fans can seemingly count on their series standard uptake to inform them on the argument that these objects will make no matter which game they play.

“Seemingly,” as it should be noted that these arguments, and the ways they are taken up through series standard uptake, are most consistent in 2D *Mario* games such as *SMB* and *SMW*. As has been discussed in previous chapters, *SM64* represented a step forward for not just *Mario*, but gaming in general, being one of the first games of its kind. With that in mind, what happens when the concepts of power-ups and blocks are transferred into the third dimension with *SM64*? Did the arguments remain the same as they were re-imagined for use in a 3D platforming game, or were they rewritten entirely? And in either case, what seems to be the rationale behind the way these arguments are presented, and in what way is the arguments’ structure influenced by the concept of series standard uptake?

The answer to the first question is a bit of both. *SM64*’s power-ups, taken as PRSes, convey an entirely different argument when compared to the power-ups in the 2D *Mario* games. *SM64* completely lacks the super mushroom, cape feather, or any other power-ups that enhance Mario’s

standard suite of abilities, shifting the dynamic between the player and these items. Where 2D power-ups argued their worth as enhancements to normal gameplay, *SM64*'s power-ups argue that they are only meant to be used in certain places for certain purposes. Examples include the metal cap, which turns Mario's body into a dense alloy. It is meant to be used to traverse areas filled with poisonous gas and to sink to the bottom of bodies of water. While the player can try to use it in other areas if they wish, the cap is on a strict timer, and it only appears near the areas where it is meant to be used, effectively chaining it to that use.

Despite the almost entirely different argument surrounding the use of power-ups in *SM64*, the means for acquiring them retains some of the same PRSes. *SM64* lacks question blocks, meaning that the element of randomness is gone. However, the game does have colored blocks, which bear a strong visual resemblance to the ones in *SMW*. In addition, much like in *SMW*, when *SM64*'s blocks are first encountered, they appear as unfilled, dotted outlines. Naturally, players can only fill these outlines by finding the corresponding switch (Figure 23).



Figure 23. Like in *SMW*, *SM64*'s colored blocks start out unfilled.

Multiple PRSes from the 2D games are thus in effect in *SM64*, including the argument that jumping is useful for busting open boxes that may contain enhancements to one's character. However, while this important part of the block and power-up argument is still in *SM64*, most other arguments related to blocks are gone. Aside from dropping the randomness element, *SM64* also drops the use of blocks as bridges to new areas seen in *SMW*. Since *SM64*'s power-ups tend

to be placed only near areas they are to be used, blocks are almost always encountered alone, eliminating the chance for multi-block arrangements like in *SMB* and *SMW* (Figure 24).



Figure 24. A lonely colored block in *SM64*. Unlike in the 2D *Mario* games, *SM64* rarely arranges multiple blocks near each other.

With the above in mind, there seems to be no reason for *SM64* to keep the block PRS, especially considering the more context-sensitive usage of power-ups. It would seem that the power-ups could have simply been left out in the open without creating much of a difference. This conclusion is supported by future 3D *Mario* games not using blocks to contain power-ups; *Super Mario Galaxy*, for example, places power-ups inside bubbles the player simply walks into. Considering the elements of randomness and bridge-building are gone, it seems that blocks are, from a game design standpoint, totally unneeded.

Thus, the question of rationale on the game designer's part returns here. The question has no answer, but there are, as always, several possibilities. One is that the designers left blocks in *SM64* because they were aware of their status as a series PRS. This decision would thus be based on the expectation that *Mario* veterans would understand the function of blocks even if seeing them for the very first time in 3D rather than 2D.

This may have been further motivated by the need to ease players of the 2D games into the 3D realm. Given that *SM64* was the first 3D game in the series, it may have been intimidating to

returning players. The developers may have given them the familiar colored block to argue that the rules were still similar to the 2D games through the use of a familiar and, by that point, iconic piece of imagery. While returning players will ultimately find the argument has changed, the use of the block PRS still serves its purpose of conveying the *SM64*'s rules to them.

Perhaps the smart, iterative use of series PRSes like the colored block is part of why *SM64* was such a successful game, and how it manages to mostly avoid giving players moments of frustration like those encountered by Arin. The ways that series PRSes can be used leave many questions open that are ripe for future research as well. How do PRSes evolve as a result of using and gradually transforming familiar series PRSes? Do developers feel obligated to include such PRSes thanks to knowledge of how important they are to their players? Do they feel “trapped” by the series conventions that invariably arise because of series PRSes? And on the player side, what determines a series PRS being subverted as frustrating? How many games must a player either play or be aware of before they form a solid series standard uptake lens? And perhaps most interestingly, how does knowledge gained through application of a series standard uptake lens influence one's perception through other lenses, such as modder uptake?

Implications of Modder Uptake - PRSes and Assets in Mashup Hacks

Kaze Emanuar is the online handle of a modder who is well-known for doing some truly outlandish and impressive things with *SM64* ROM hacks. Emanuar is most famous for hacks which take elements from games other than *SM64* and “mash them up” with the N64 game, recreating them in *SM64*'s game engine. One example of this is *Super Mario Bros. 64 (SMB64)*. As its title implies, it tries to combine the original *SMB* with *SM64*, recreating everything - levels, power-ups, etc. - that was in *SMB* in *SM64*'s game engine (Figure 25).



Figure 25. *SMB64* vs. *SMB*. The ROM hack tries to recreate every element of the original, including the UI, the level design, and even the 2D perspective.

Thinking in terms of PRSes and uptake lenses, it might be tempting to assume *SMB64* is, as its name implies, an amalgam of PRSes from *SMB* and *SM64*. This would, on the surface, make sense given the premise of modder uptake, which lets modders view PRSes as transferable, and would thus seemingly allow Emanuar to effortlessly transfer arguments from *SMB* to *SM64*. Yet in the case of *SMB64* (and other mashups), the PRS situation is nowhere near that clear-cut. Complications arise as a result of connections between procedural arguments and game assets, the often unknowable nature of the intentions motivating those who make use of modder uptake, and the unpredictability of player uptake toward mods. These complications raise questions about the nature of PRSes and their relationship to uptake lenses, modder certainly among them.

To begin with, a “game asset” simply refers to any individually coded piece of a game. This includes enemies, environments, power-ups, and even specific elements like *SM64*’s doors. Every asset in any game makes a procedural argument, and on a meticulously analytical level, these arguments are situated alongside others that give them their specific contextual meaning. *SM64*’s doors couldn’t make their argument if Mario was given the ability to walk through walls, as the PRS originating from Mario’s traversal capability would cancel out the PRS of the door. Of course, this view of PRSes assumes the game is viewed through the lens of standard uptake, as viewing the game with other lenses (such as speedrun) would invalidate the door regardless. This

view of PRSes assumes the same of a ROM hack like *SMB64* as well, given that there is clearly a way they are meant to be played, especially with how they use assets specifically to echo rules and PRSes from multiple games.

The connection between game assets and PRSes is what makes *SMB64* so intriguing. From a glance, it may look like the hack just rebuilds *SMB* using *SM64*'s assets. In some ways, this is true. The levels and enemy placements are accurate, the game is set in 2D, and generally, Emanuar's intention seems to be giving players the chance to play *SMB* with *SM64*'s mechanics, effectively gathering the skin and organs of *SM64* and grafting them over the skeleton of *SMB*. This required Emanuar to choose assets from *SM64* that are the equivalent to assets in *SMB*, since there is obviously no way to transpose assets from a 2D NES game into a 3D N64 game. One asset that made the transition with minimal major alterations is the Goomba. In both games, it is the basic cannon fodder enemy that resembles a walking mushroom (visible in Figure 25), though it functions a bit differently in either game. In *SMB*, it walks forward endlessly; in *SM64*, it patrols a small area until the player approaches it, at which point it will charge them.

SMB64 uses the *SM64* version of the Goomba, brought over almost without any changes. This causes the Goomba to make an argument that is overall more similar to *SM64* than *SMB*. *SMB*'s Goombas are generally more "aggressive," arguing to the player that they have to jump out of their way or suffer a hit. *SM64*'s Goombas are generally more "passive," and in *SMB64*, they definitely make this argument in part, as the player can avoid antagonizing them entirely. However, while the code of the Goomba asset is the same and is being run in its intended engine, the argument it conveys is nonetheless transformed thanks to the altered perspective. In *SM64*, Goombas are avoided more easily since the player moves with full three-dimensional freedom. In *SMB64*, the player can only move forward and backward on the ground. This restriction, coupled

with the Goombas' argument of passivity, forces the player to approach them differently, using precise platforming over floating objects if they wish to avoid entering their patrol zones. This is just one example of how by changing one asset, other assets around it are changed too. For games, procedural arguments require all pieces to work according to the intended rules. Without all those pieces in place, a different argument will arise, and the idea that PRSes can be mingled so easily between games seemingly falls apart.

Of course, it's very possible that Emanuar was aware of this and knew it was inevitable. *SM64* can't be *SMB* without losing part of itself in transition, just as the inverse would be true. Emanuar successfully offers the experience of playing *SMB* through the game engine of *SM64*, but "mashing together" two very different games creates an end result that is somewhat messy, and the longer it is looked at, the less it seems to resemble either of the pieces that it began as. However, with Emanuar's intentions set on the goal of recreating *SMB* as authentically as he can, he did not give up on using assets which try to present *SMB64* as two sets of arguments.

Emanuar's usage of assets lifted directly from *SM64* falls in line with modder uptake, which designs mods for players and anticipates how those players will take up the mod's PRSes. The fact that the Goomba, while crammed into a 2D corridor, still works as it did in *SM64*, serves as evidence of this. He seemingly kept the Goomba close to how it originally functioned because he was anticipating players would bring in standard uptake lenses developed for *SM64*. Those players would presumably expect a Goomba in the *SM64* engine to behave a certain way, and *SMB64* mostly fulfills those expectations when it argues that Goombas are primarily passive. Of course, as has been discussed above, the Goomba is not identical to how it would be in *SM64*. Yet given that it is constrained to a 2D plane, it functions closely *enough* to how players would expect based on their standard uptake of *SM64* to where most of them would likely accept it.

The same could be said of other *SM64* assets that *SMB64* imports without major changes. However, as was alluded to, Emanuar would have been more than capable of reprogramming Goombas to behave aggressively as they do in *SMB*. He seemingly did not for the above reasons, yet in order to create an experience that plays like *SMB* and *SM64* are merged into one game, Emanuar cannot rely exclusively on unchanged *SM64* assets. While plenty of equivalents exist, there are some *SMB* assets that lack analogues in *SM64* but are crucial to *SMB*'s design.

In these instances, Emanuar would've had to do some deeper tinkering with *SM64*'s code. In some cases, this entailed creating his own assets totally from scratch to ensure that the *SMB* experience is provided in its entirety. Assets that were made by Emanuar seem to be designed with the intent of appealing to standard uptake of *SMB*. However, the inevitable differences from the originals result in a PRS that is unlike the original and takes on some unexpected uptakes, while also bringing up the question of where the PRS came from based on the intent behind it.

A good example is the fire flower. *SM64* completely lacks an equivalent to this power-up, so naturally, Emanuar created his own to function as similarly to the *SMB* version as it could. *SMB64*'s fire flower turns Mario's clothes red and white, allows him to sustain an extra hit, and, of course, allows him to throw fireballs at his foes. Similarly to how use of mostly-unchanged *SM64* assets seems intended to get players to think in terms of their *SM64* uptake lens, the use of a custom-programmed fire flower seems meant to get players to think of their *SMB* uptake lens. However, as alluded to above, the fire flower in *SMB64* inevitably functions differently from the original. As a result, it conveys its own unique procedural argument to the player, suggesting that some amount of standard uptake needs to be developed for the mod itself, rather than trying to exclusively use standard uptake lenses for *SM64* and especially *SMB*.

This can be seen when watching Darby, creator of the channel Blue Television Games. Back in 2016, Darby uploaded a playthrough of *SMB*, a game that he is intimately familiar with. His standard uptake lens for *SMB* is on display throughout the playthrough, including a moment during World 1-1 where he obtains a fire flower. Upon obtaining it, he recognizes exactly how the game argues it is meant to be used and tosses fireballs at enemies. The fireballs bounce along the ground in a pre-programmed arc, never going any higher or lower as they travel forward. When the fireballs hit a wall, they instantly dissipate. This aspect of the fire flower argues a limitation Darby clearly recognizes, as he never attempts to use the fire flower to attack an enemy that is on the other side of a wall. The same can be seen with how he interacts with *SMB*'s other PRSes as well, as he never tries to act outside the limitations that the game argues for him (Blue Television Games, 2016).

Years later, in 2018, Darby posted a playthrough of *SMB64*. During this playthrough, Darby encounters several PRSes that he seemingly tries to apply his standard uptake of *SMB* to, only to find that they convey some different arguments as to their capabilities or functionality. The fire flower is one of them, of course. Upon acquiring the item, he tries using it in a similar way to how he would use it in *SMB*. Yet the fireballs he throws move not in a repeating bounce, but in a straight line, similar to a baseball pitch. Their arc is determined by physics simulations in the *SM64* game engine, which causes them to bounce around unpredictably. They do not immediately disappear upon hitting walls, and in some cases, they actually “crawl” up and over those walls, allowing them to hit enemies on the other side. Darby takes advantage of all these unique qualities and uses the *SMB64* fire flower in ways he would never use the *SMB* version. The argument posed by the power-up in the hack is obviously different from the one that it poses in the original game,

and it seems that as a result of this, Darby ceases to rely on his standard uptake of *SMB* and relies instead on standard uptake of *SMB64* (2018).

If Emanuar's intention was to draw upon players' standard uptake lens gained from *SMB*, he succeeded in some ways and failed in others. The fire flower in *SMB64* conveys some of the same arguments as the one in *SMB* - namely, it functions as a projectile that defeats enemies. However, it also conveys arguments that inform the player of capabilities far beyond those of the original fire flower, such as its extended persistence after being thrown and the fact that it can bounce in myriad directions. It cannot be said for certain whether Emanuar intended for the *SMB64* fire flower to convey all these arguments, but it is clear that the asset isn't a one-to-one copy of the one from *SMB*. Despite the similarities that are indeed present, Emanuar's creation is seemingly original on his part.

The creation of original assets that resemble those in another game for mods raises numerous questions about modder uptake. Are PRSes modder uptake makes use of determined by the intentions of the modder? Where does one draw the line between a PRS that is meant to resemble one from another game and a PRS that is mostly original and created by the modder? What does the ability to "perceive" PRSes as transferable truly mean in the case of mods like *SMB64* where the assets behind those PRSes have to be remade? Where should the line be drawn between the physical code of a game asset and the PRS that it functions as? And perhaps most importantly of all, what does it mean to develop a standard uptake lens for a mod? How much of it is influenced by one's expectations based on the game of basis, or in this case of adaptation, before the mod becomes its own argumentative entity? Whatever the case may be, it is clear that modder uptake can become extremely complex and messy in a hurry, doubtlessly requiring more research before it can even begin to be understood.

Self-Imposed Challenge - Uptake Lens for Future Research

I've invoked the image of Bogost's daughter stepping over cracks on a path a few times, and I'm going to do it again here because if I had to boil my thesis down to just a single concept, that would definitely be it. Games argue to their players about how they are meant to be played, just as a sidewalk might argue to pedestrians about how it is meant to be walked on. However, the recipient of persuasion always has a choice in how they take up argument for intended use. They could walk along the path and not worry about whether they step on the grout-filled cracks, or they could choose to challenge themselves by only stepping over those very same cracks. They have the ability to experiment and "play" with whatever they encounter however they wish, and that includes video games just as much as footpaths.

Relatedly, an uptake lens I regret not exploring is the self-imposed challenge, or SIC. That this lens is named as such suggests that it isn't the result of players pushing against the limitations of games, such as with speedrun uptake or modder uptake. Rather, this lens is named because it is based on players imposing limitations on themselves. The motivation for this uptake lens is usually simple, revolving around players' desires to challenge themselves, oftentimes with a game that they have played extensively and feel they want to breathe new life into.

SIC uptake is also interesting because it does not necessarily take up PRSes in such a way that leads to players playing the game in a way that differs from how it is meant to be played. Rather, it follows the rules as they are set out, but applies additional strictures to make the game more challenging than it is likely intended to be. An example of this might be playing through *SMB* without using any power-ups. Power-ups are a PRS that argues to the player that they can strike blocks to possibly find an enhancement to their normal abilities. They are meant to make

the game easier, and by choosing not to use them, the player might be disagreeing with a certain argument the game is putting forth.

“Might be,” because in the case of power-ups and other optional items in games that can be rejected within the confines of the normal rules, it is somewhat difficult to say whether a self-imposed challenge constitutes a proper uptake lens. Choosing not to use power-ups could be something someone playing the game with a standard uptake lens would do. At that point, SIC might seem to be a subcategory of standard uptake and nothing more. Whether developers intend for players to put limits on themselves can also be difficult to determine. *Dark Souls*, for example, prides itself on difficulty in its marketing. One of the character classes that a player can select, the Deprived, is deliberately designed to be weak in order to pose a stiffer challenge.

There are, however, more extreme pieces of evidence from SICs which offer greater support for SIC uptake being a distinct lens. Such evidence takes the form of challenges which, similar to modder or speedrun uptake, often require a game to be played using external means. These sorts of SICs are often designed by their creators to be incredibly difficult, requiring elaborate means for resolving them which simply do not exist in the regular game. *SM64* has a few challenges like this, all of which can be traced back to a specific PRS.

One example of such a challenge is the A button. The A button, on an N64 controller, allows Mario to jump in *SM64* when it is pressed. Extending procedural rhetoric to the physical, the “A press” could be considered a PRS. The player, upon finding that they can make Mario jump by pressing A, will likely take up an argument that they will need to jump to beat the game. Or, to be more precise, they will need to press the A button.

Youtuber Pannenkoek2012 (Pan) is one of the most knowledgeable people on *SM64*. Fascinated with the game from a young age, he mastered it to an extent that playing “normally,”

even down to the controls, was no longer a challenge. Because of this, he created one of his own - to get every star in *SM64* without using an A press. This is an extremely difficult challenge. Being a platforming game, *SM64* is designed around the expectation the player will be jumping around all the time. Beyond that, the A button is also used for the most famous glitch in *SM64*, the BLJ, which would make some stars easier to get to if the only limitation was that the player could not use normal jumps. Yet Pan's challenge states that using that button on the controller isn't permitted. Defying the procedural argument of the A button, how can he hope to collect every star?

Pan's answer has been to tackle *SM64* piecemeal, figuring out how to eliminate the need for each A press until the game can definitively be beaten without them. This has entailed going after each star in the game as something akin to a puzzle, with perhaps the most famous example demonstrated his video called "Watch for Rolling Rocks - 0.5x A Presses." This video sees him get the star called "Watch for Rolling Rocks" in the level Hazy Maze Cave without using a full A press, instead using only half of one. Half an A press is explained by Pan as holding down the A button before entering a level, which creates a constant input that Mario can take advantage of to use moves that would otherwise be impossible, but it does not qualify as a full A press (2016).

Pan needs half an A press, as Hazy Maze Cave is a complex level full of uneven terrain. Getting through without an A press is not only hard, but technically impossible for most players. Pan does it mainly by utilizing parallel universes, or PUs as he calls them. PUs, in simple terms, are copies of the game's collision data that exist outside the boundaries of the game's levels. They normally cannot be accessed unless certain glitches are used - and even with a full A press, these glitches are almost impossible for a human to execute, the reason being that they require hours upon hours of real-world time (Figure 26).



Figure 26. Pan's PU map and multi-display view from his video.

The first of these glitches alone necessitates that Mario build up speed for twelve hours. The amount of speed that is gained enables Mario to pass through a wall and end up in a PU. By jumping from one PU to another (which, again, is grossly oversimplifying), Pan is able to move Mario through the level without using a full A press. However, the means behind Pan's SIC are not as important discussing it as an uptake lens, but rather, the massive time commitment. Pan had to make use of an emulator and various external tools. Mario's movements are controlled not by Pan's fingers, but by an outside software he programmed to do what it needed to. The in-game clock can be sped up with such software, meaning that the multi-hour wait for Mario's speed can be condensed. This clearly indicates a version of the SIC uptake lens that cannot be accomplished in the game normally, providing more evidence for this lens being distinct from standard uptake.

Separating Pan's challenge from standard uptake even further, I would argue that it, alongside any other SIC, is effectively a much more extreme and complex version of Bogost's daughter stepping over cracks. In either case, the more obvious and seemingly efficient means of using a system - *SM64* or a sidewalk - is ignored in the name of taking on a greater challenge. There needs to be some kind of system making a specific argument for how it should be interacted with, an argument of limitation that will be mostly accepted by the majority of users. Yet there seems to always be those who either take limitations further, loosen those limitations, or do some combination of both in the interest of creation. There is always a desire to take things further, to

be investigative, and to create challenges that end up teaching players and people more about the games they play and perhaps the world around them.

This is why I find PRSes and uptake lenses so attractive for further research. They are, from a video gaming and digital rhetoric perspective, what we do all the time as human beings. We push against these boundaries not because there is immediate benefit or it seems “necessary,” but because of the potential that lies beyond the unknown, the improvement of intellect and skills that necessarily comes from taking on such challenges. Speedrunners, modders, and those who take on self-imposed challenges in video games are doing something that most people do not, applying considerable cognitive effort to the act of interfacing with systems for their own sake, and through their self-betterment and digital discoveries, for the sake of those around them. There’s no other reason that speedruns draw in thousands of viewers, or that Pan’s *Rolling Rocks* video has over three million views. Systems may make arguments as to how they should be used, but people’s curiosity will always lead them to finding something more.

REFERENCES

- Adams, Douglas. (1998). *Starship Titanic* [Software]. Digital Village; Simon and Shuster Interactive. Microsoft Windows. Video Game.
- Alaris_Villain. (2017). "SM64 Categories and Glossary." *Google Docs*. Web. Accessed March 20, 2019.
<https://www.speedrun.com/sm64/guide/oilwj>
https://docs.google.com/document/d/1DMFyIXIFNERY_kPsFeGI9GhtyKz8sXE4zaJtX9MUv2g/edit?hl=en&pli=1#
- Bandy, Nathaniel. (2016). "Super Mario Star Road - Nathaniel Bandy." *YouTube*, uploaded by Nathaniel Bandy. Web. Accessed May 26, 2019.
<https://www.youtube.com/watch?v=SsBHglBmH-Q>
- Benzies, Leslie. (2004). "Grand Theft Auto: San Andreas" [Software]. Rockstar Games: Rockstar North. Edinburgh, Scotland. Video Game.
- Bogost, Ian. (2005) "The Rhetoric of Exergaming." The Georgia Institute of Technology. 686 Cherry St. Atlanta, GA 30332 USA. Web.
<http://bogost.com/downloads/i.%20boogst%20the%20rhetoric%20of%20exergaming.pdf>
- Bogost, Ian. (2007) "Persuasive Games: The Expressive Power of Video Games." Cambridge: The MIT Press. Cambridge, Massachusetts. Print. PG. 2-3, 42-43.
- Bogost, Ian. (2008). "The Rhetoric of Video Games." *The Ecology of Games: Connecting Youth, Games, and Learning*. Edited by Katie Salen. The John D. and Catherine T. MacArthur Foundation Series on Digital Media and Learning. Cambridge, MA: The MIT Press, 117–140.

- Bogost, Ian; Ferarri, Simon; Schweizer, Bobby. (2010). "Newsgames: Journalism at Play." MIT Press. Cambridge, Massachusetts. Print.
- Bogost, Ian. (2016). "Play Anything: The Pleasure of Limits, the Uses of Boredom, and the Secret of Games." Basic Books. Print.
- Buchanan, Scott. (2016). "SM64 - Watch for Rolling Rocks - 0.5x A Presses (Commentated)." *YouTube*, uploaded to pannenkoek2012. Web. Accessed June 22, 2019.
- Cozic, Laurent; et al. (2004). "Intuitive Interaction and Expressive Cinematography in Video Games". Middlesex University. Lansdowne Centre for Electronic Arts.
- Chen, Jui-Hung; Chang, Han-Bin; Shen, Chun-Yi; Wang, Te-Hua; Chang, Wen-Chih. (2009). "Developing the immersive learning environment by integrating the interactive video and ubiquitous technologies." Proceedings of the first ACM international workshop on Multimedia technologies for distance learning. Beijing, China.
- Colby, Richard. (2014) "Writing and Assessing Procedural Rhetoric in Student-Produced Video Games." *Computers & Composition*, vol. 31, pp. 43–52.
- Cogndon, Benjamin. (2018). "AGDQ 2018 Stats." GDQ Stat. Web. Accessed January 5, 2019. <https://gdqstat.us/previous-events/agdq-2018/?series=0>
- Cruz-Cunha, Maria Manuela. (2012). "What is Win-State." *Handbook of Research on Serious Games as Educational, Business and Research Tools (2 Volumes)*. IGI Global. Polytechnic Institute of Cavado and Ave. Print. PG. 278.
- CVG Staff. (2006). "PC Interview: Half-Life 2 Minerva Mod." *Computer and Video Games Magazine*. Web. Accessed March 18, 2019, through web.archive. <https://web.archive.org/web/20071016121618/http://computerandvideogames.com/article.php?id=144068+min>

- Darby. (2016). "Darby and Lauralee Play Super Mario Bros. - NES - Part 1." *YouTube*, uploaded to Blue Television Games. Web. Accessed June 18, 2019.
<https://www.youtube.com/watch?v=bQn0kuN1III&list=PLjA9gF9YbBHwmWtmy9TS1ctSbHoDIBGD7>
- Darby. (2018). "Super Mario Bros. Remade in Super Mario 64 | Part 1." *YouTube*, uploaded to Blue Television Games. Web. Accessed June 18, 2019.
<https://www.youtube.com/watch?v=iZhauEugsx0>
- DeLeon, Chris. (2013). "Rules in Computer Games Compared to Rules in Traditional Games." *YouTube*. Uploaded by Chris DeLeon of Gamkedo. Web. Accessed February 2, 2019.
<https://www.youtube.com/watch?v=5n-apftKNVk>
- DGR. (2018). "SUPER MARIO STAR ROAD (Mario 64 Hack) | Blind Playthrough." *YouTube*, uploaded by DGR. Web. Accessed May 28, 2019.
https://www.youtube.com/watch?v=FKS_gDvHVXk
- Dutchuk, Mark; Muhammadi, Khalid Aziz; Lin, Fuhua. (2009). "QuizMAster A Multi-Agent Game-Style Learning Activity, Proceedings of the 4th International Conference on E-Learning and Games: Learning by Playing." *Game-based Education System Design and Development*. Banff, Alberta, Canada.
<https://dl.acm.org/citation.cfm?id=1615337>
- Ferrara, James. "Games for Persuasion: Argumentation, Procedurality, and the Lie of Gamification." *Games and Culture*. Vol 8, Issue 4, pp. 289 - 304. August 27, 2013. Web.
<http://journals.sagepub.com/doi/full/10.1177/1555412013496891#articleCitationDownloadContainer>

- Ferrari, Simon. "The judgment of procedural rhetoric." Georgia Tech Theses and Dissertations [20837]. School of Literature, Media, and Communication Theses and Dissertations April 8, 2010. <https://smartech.gatech.edu/handle/1853/33915>
- Fine, Benny; Fine, Rafi. (2018). "TEENS PLAY SUPER MARIO 64 - Part 1 (React: Twitch Let's Plays)." *YouTube*. Uploaded by FBE Live.
https://www.youtube.com/watch?v=-Pjyw_y5MLg
- Foster, Adam. (2005). "MINERVA: Metastasis" [Software Modification]. Microsoft Windows. Video Game.
- Foster, Adam. (2006). "Minerva." Interview by Bob Watson. Idle Thumbs. Web. Accessed May 16, 2019.
<http://old.idlethumbs.net/display.php?id=224>
- Gee, James. (2003). What Video Games Have to Teach Us About Learning and Literacy. *Computers in Entertainment*. 1. 20. 10.1145/950566.950595.
https://www.researchgate.net/publication/220686314_What_Video_Games_Have_to_Teach_Us_About_Learning_and_Literacy
- Hanson, Arin. (2013). "Super Mario Sunshine: Double Demon Action - PART 42." *YouTube*, uploaded by Game Grumps. Web. Accessed May 22, 2019.
https://www.youtube.com/watch?v=EprNveErkqM&list=PLRQGRBgN_Enq2ZCxRmlhO0QGHqEF7VnZ&index=42
- Harm Prins. (2017). "Super Mario 64 120 star Speedrun World Record in 1:41:12." *YouTube* Video Comment. Web. Accessed April 14, 2019.
<https://www.youtube.com/watch?v=5SVLIRthpNc>

Howlongtobeat. (2019). "Super Mario 64." Howlongtobeat. Web. Accessed March 20, 2019.

<https://howlongtobeat.com/game.php?id=9364>

Harper, Todd. "Rules, Rhetoric, and Genre: Procedural Rhetoric in Persona 3." *Games and Culture*. Vol 6, Issue 5, pp. 395 - 413, March 27, 2011. Web.

<http://journals.sagepub.com/doi/pdf/10.1177/1555412011402675>

Heale R, Twycross. "What is a case study?" *Evidence-Based Nursing* 2018; 21:7-8.

<https://ebn.bmj.com/content/21/1/7>

Hidayah, Nurul; Zain, Mat; Hanis, Fariza; Razak, Abdul; Jaafar, Azizah; Zulkipli, Mohd Firdaus.

"Eye tracking in educational games environment: evaluating user interface design through eye tracking patterns." *Proceedings of the Second international conference on Visual informatics: sustaining research and innovations*, November 09-11, 2011, Selangor, Malaysia.

<https://dl.acm.org/citation.cfm?id=2074678>

hotnintendo. (2009). "How to add cheats to Project64." *YouTube*. Uploaded by hotnintendo. Web. Accessed February 2, 2019.

https://www.youtube.com/watch?v=Pc1C5BNy_z8

Kærlev, Jonas. (2013). "A Hat in Time." [Software]. *Gears for Breakfast*. Microsoft Windows. Video Game.

Kaze Emanuar. (2018). "Super Mario Bros. 64" [Software Modification]. Microsoft Windows.

Klabbers, Jan H.G. "Tensions Between Meaning Construction and Persuasion in Games." *Game Studies*. volume 11 issue 2. May 2011. ISSN:1604-7982

http://gamestudies.org/1102/articles/klabbers_book_review

- Koizumi, Yoshiaki; Usui, Kenta. (2002). "Super Mario Sunshine" [Software]. Nintendo GameCube. Nintendo EAD, Nintendo.
- Koizumi, Yoshiaki; Miyamoto, Shigeru; Shimizu, Takao. (2007). "Super Mario Galaxy" [Software]. Nintendo Wii. Nintendo EAD Tokyo, Nintendo.
- La Molleindustria. (2006). "McDonald's Video Game." [Software]. La Molleindustria: Italy. Video game.
- Lave, Jean. (2010). Teaching, as Learning, in Practice. *Mind, Culture, and Activity*. 3. 149-164. 10.1207/s15327884mca0303_2. https://www.researchgate.net/publication/261594107_Teaching_as_Learning_in_Practice
- Lee, Joel. (2015). "Speedrunning: The Future of Single-Player Games." *MakeUseOf*. Web. Accessed January 5, 2019. <https://www.makeuseof.com/tag/speedrunning-culture-future-single-player-gaming/>
- Mackey, Margaret. (2007). *Literacies Across Media: Playing the Text*. London and New York: Routledge. Print.
- Matheson, Calum. "Procedural Rhetoric Beyond Persuasion: First Strike and the Compulsion to Repeat." *Games and Culture*. Vol 10, Issue 5, pp. 463 - 480. December 31, 2014. Web. <http://journals.sagepub.com/doi/full/10.1177/1555412014565642#articleCitationDownloadContainer>
- maxaura. (2012). "is blj's banned in 70 star." Web Forum Post. Web. Accessed March 20, 2019. <https://www.speedrun.com/sm64/thread/19122/1#9nf62>
- Miyamoto, Shigeru; Tezuka, Takashi. (1986). "Super Mario Bros" [Software]. Nintendo Creative Department. Nintendo Entertainment System. Video Game.

- Miyamoto, Shigeru; Koizumi, Yoshiaki; Tezuka, Takashi. (1996). "Super Mario 64" [Software].
Nintendo EAD. Nintendo 64. Video Game.
- Miyazaki, Hidetaka. (2011). "Dark Souls" [Software]. FromSoftware: Bandai Namco Games.
PlayStation 3, Xbox 360, Microsoft Windows, PlayStation 4, Xbox One, Nintendo
Switch.
- Nilsen, Martin. (2019). "Xplorer 64 - The Ultimate Cheat Cartridge." NESWORLD.com.
NESWORLD. Web. Accessed November 20, 2018.
<http://www.nesworld.com/article.php?system=n64&data=n64-xploder>
- Nintendo of America. (1996). "Super Mario 64 Manual." Nintendo of America. Print.
- O'Malley, James (2015). "30 Best-Selling Super Mario Games of All Time on the
Plumber's 30th Birthday". Gizmodo. Univision Communications. Web. Accessed
November 24, 2018.
- OmegaEdge29. (2009). "Kaizo Mario 64" [Software Modification]. Microsoft Windows.
- Paul, Christopher A. "Optimizing Play: How Theorycraft Changes Gameplay and Design."
Game Studies. Volume 11 issue 2. May 2011. ISSN:1604-7982
<http://gamestudies.org/1102/articles/paul>
- n.a. (2019). "Progression Gameplay." Technopedia. Technopedia Incorporated. Web. Accessed
October 15, 2018. Web.
<https://www.techopedia.com/definition/27044/progression-gameplay>
- ROMhacking.net. (2019). "Dictionary of ROMhacking Terms." ROMhacking.net. Web.
Accessed May 15, 2019.
<http://www.romhacking.net/hacks/4599/>

seeker___. (2018). "Doom (2016) Finished In a Staggering 28 Minutes." Uploaded by IGN. Web. Accessed February 12, 2019.

https://www.youtube.com/watch?v=3x_Toq-KrNc

Sezen, Tonguc Ibrahim; Isikoglu, Digdem (2007). "FROM OZANS TO GOD-MODES: CHEATING IN INTERACTIVE ENTERTAINMENT FROM DIFFERENT CULTURES" (PDF): 8. Retrieved 2009-01-24.

Skelux. (2011). "SM64 Star Road - Sandy Slide Secret (Major Hack Preview)." *YouTube*, uploaded by Skelux. Web. Accessed May 30, 2019.

<https://www.youtube.com/watch?v=CRLGg6vg50s>

SimpleFlips. (2015). "Star Road; 0 Star World Record (5:09)." *YouTube*, uploaded by SimpleFlips. Web. Accessed May 30, 2019.

<https://www.youtube.com/watch?v=3LkitpKyFQA>

Sicart, Miguel. (2008). "Newsgames: Theory and Design." In: Stevens S.M., Saldamarco S.J. (eds) Entertainment Computing - ICEC 2008. International Conference on Entertainment Computing. ICEC 2008: Entertainment Computing - ICEC 2008. Lecture Notes in Computer Science, vol 5309. Springer, Berlin, Heidelberg pp 27-33. Web.

https://link.springer.com/chapter/10.1007/978-3-540-89222-9_4#citeas

Sicart, Miguel. (2011). "Against Procedurality." *Game Studies*. Volume 11 Issue 3. ISSN:1604-7982

http://gamestudies.org/1103/articles/sicart_ap

Skelux. (2011). "Super Mario Star Road" [Software Modification]. Microsoft Windows.

SpeedDemosArchive. (2014). "Getting Started." SpeedDemosArchive. Web. Accessed March 20, 2019.

https://kb.speeddemosarchive.com/Getting_Started

SpeedRunsLive. (2019). "FAQ." SpeedRunsLive. Web. Accessed March 20, 2019.

<http://www.speedrunslive.com/faq/>

Speedrun.com. (2019). "Super Mario 64." Speedrun.com. Web. Accessed March 20, 2019.

https://www.speedrun.com/sm64#0_Star

SullyPwnz. (2017). "Super Mario Star Road: Multiplayer - Part 1: The Revenge of King Bob-omb! (2 Player)." *YouTube*, uploaded by SullyPwnz. Web. Accessed May 30, 2019.

<https://www.youtube.com/watch?v=IOItS-zzQC0>

Sweester, Penelope, et al. "Gameflow in Different Game Genres and Platforms." *Computers in Entertainment*, vol. 15, no. 3, 15 Feb. 2017, pp. 1-24.

Tezuka, Takashi; Miyamoto, Shigeru. (1990). "Super Mario World" [Software]. Super Nintendo Entertainment System. Nintendo EAD, Nintendo.

TTDLX. (2011). "Super Mario 64 Full Playthrough (120 Stars + Yoshi bonus area)." *YouTube*.

Uploaded by TTDLX. Web. Accessed January 18, 2019.

<https://www.youtube.com/watch?v=XnqGFn0Byg4>

Valve Corporation. (2004). "Half-Life 2" [Software]. Valve Corporation. Microsoft Windows. Video Game.

Vinny. (2017). "[Vinesauce] Vinny - Super Mario 64 (part 1)." *YouTube*. Uploaded by Vinesauce: The Full Sauce. Web. Accessed January 18, 2019.

https://www.youtube.com/watch?v=Pshc_KWq0Uo

Wirtanen, Josh. (2017) "The 3D Platformer: How 1996 Witnessed the Birth of a Genre."

Retrovolve. Web. Accessed November 22, 2018.

Whalen, Ernie. (2018). "Super Mario 64 - 120 Star Run - *Blind Playthrough." *YouTube*.

Uploaded by Flapper.

<https://www.youtube.com/watch?v=56TnQjzaOlc>

Ukikipedia. (2019). "Backwards Long Jump." Ukikipedia. Web. Accessed April 20, 2019.

https://ukikipedia.net/wiki/Backwards_Long_Jump

Xiah. (2015). "Super Mario 64 - 0 Star Speedrun in 6'47"64." *YouTube*, uploaded by Xiah7s.

Online Video. Accessed April 22, 2019.

<https://www.youtube.com/watch?v=Itq1jdG01O8>