Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc

Indicators in *Super Mario Maker 2*: Evolution and Rhetorical Signification in Visual Languages

Nathan W. Eloe[a], Trevor C. Meyer[b]

[a]*School of Computer Science and Information Systems, Northwest Missouri State University, USA*
[b]*Department of Language, Literature, and Writing, Northwest Missouri State University, USA*

ARTICLE INFO

Article History:
Submitted 7.31.2022
Revised 9.1.2022
Accepted 9.6.2022

Keywords:
Visual Languages
Video Games
Usability and Accessibility
Linguistics
Metadiscourse
Abusive Game Design

ABSTRACT

As a key element of user experience, communicating expectations to a user through visual elements instead of natural language can produce a more intuitive interface as users interact with a system. While such visual languages are developed by domain experts for specific purposes, these languages can also grow, change, and evolve within a community of users in the same manner as natural language. Video games are one area where communication with the user directly affects the enjoyability, usability, and accessibility of the system. While work has been done to create and leverage visual languages in order to gamify learning or improve accessibility, this research focuses on the creation and evolution of these visual languages by external experts. This exploration of a "crowd–sourced" context–aware visual language through the lens of a structural linguistic framework examines a system of indicators that has evolved over time, created by the very users of the language, to communicate the expectations and necessary actions to complete a task with other members of the community. Extending the analysis and design of visual languages to account for linguistic theory affords designers new tools and approaches in their own work, as too often disciplinary experts can be restricted by conventional understanding, "best practices", and what can be considered a "legitimate" object of study. In this paper, we examine a recent and widely popular visual language of indicators used in the *Super Mario Maker* games, and show how this use of indicators is central to making a usable text, a playable level, creating a relationship between player and designer that foregrounds the human elements in creating a visual language to assist users in a task. Or, in the case of "Troll" levels, prevent the user from doing so.

1. Introduction

Video games have long motivated progress in many aspects of Computer Science; algorithms (e.g. The Fast Inverse Square Root [1], often attributed to John Carmack in the implementation of *Quake*), hardware, and even education and pedagogy [2, 3] are but some areas of Computer Science that have been improved through video games. Games fulfill a particular need in entertainment; they can cater to a diverse audience while providing an interactive, not passive, experience that can be consumed by one or more players, whether separately or together.

Unlike storytelling, art, or video content alone, video games allow the player to interact with a tactile environment while processing auditory and visual information, creating a multisensory experience. This complexity has motivated a wide range of scholarship, as Johansen Quijano and Matthew Wilhelm Kapell explain in the *Introduction to The Composition of Video Games: Narrative, Aesthetics, Rhetoric, & Play*. Moreover, this complexity has also produced "a question that has been left largely unanswered: is it possible to consider video games as ludic, narrative, and rhetorical texts simultaneously?" [4]. That is, in what ways and by what methods should scholars, of any discipline, approach the composed "text," interwoven code and color, pixels and programming languages? Answers to this inquiry help provide the affordances and constraints for this paper's inquiry.

The ludic position, a sort of formalism, holds that video games should primarily be understood as compositions of actions and events, the game mechanics of "buying Baltic Avenue" or "jumping on a Koopa (a certain enemy that be defeated by jumping on its back, making a interactive Shell,

nathane@nwmissouri.edu (N.W. Eloe); tmeyer@nwmissouri.edu (T.C. Meyer)
ORCID(s):

DOI reference number: 10-18293/JVLC2022-N2-020

but it will 'revive' if enough time passes)". A contrasting position holds that game are "indeed, a narrative form and are best understood as such", and narratology, therefore, places games alongside television, film, literature, and comics as content, texts, with a beginning, middle, end, and tension motivating the player's actions [4].

Others, of course, have found nuance between the space of play and story, as both elements generally appear in most games, ranging from the (allegedly) narrative–free *Tetris*, to the more narratively driven, but still loosely, *Mario* franchise, to be explored in this paper. In the middle are games like *The Elder Scrolls*, *Mass Effect*, and *Fallout* series, among many others, which balance choice, consequence, and story in various ways, and at the narrative end is something like *Dragon's Lair*, which is a series of animated cut–scenes with the player's timing and choice determining whether the animated film will continue or the brave hero will meet his brutal demise!

Regardless of the emphasis on narrative and play from others, this paper concerns the third manner and method mentioned by Quijano and Kapell, and that pursued in their own work, studying video games as "rhetorical texts"; that is "not just as a medium, but also as visual and linguistic texts, as rhetorical devices, and as cultural artifacts", that directly communicate with, inform, and persuade readers, users, or players, in any number of (again) complex and situationally–specific ways, what we can understand as "protocol". In his book by the same name, Alexander Galloway defines *Protocol*, as "a set of recommendations and rules that outline specific technical standards" [5]. We can view these material networks not merely as instrument, nor as metaphorical communication, but as materially mediated communication, which requires a common ground for usability, specific goals and means to achieve them, which therefore produces normalizing assumptions.

Video games that present the player with a goal and expect them to reach it provide a challenge for developers to communicate to the player how to complete that posed goal. Often these games are presented as a series of levels (like the *Super Mario* series), though some use a more open world (ranging from gated progression "Metroidvania" games to fully open world games like *Skyrim*). In level–based games, each level must essentially act as an independent implementation of a visual language, providing cues for the player to determine the necessary series of actions to reach goal, as well as maintain the motivation to complete such a goal. In producing "normative" structure, these visual instructions allow the player to navigate all the various situational factors by producing limits and limiting expectations.

Through the example of the *Mario* series, we can see a long–standing and influential approach to visual language, both linguistically and aesthetically, as well as game design and gaming culture, because of *Mario*'s "prominent video game legacy that has been widely circulating since 1985" [6].

Indeed, Nintendo's own "professional game design best practices," promote a " 'user–centered' approach to game and level design" that works to prepare "players by gradually introducing and ramping up challenges", beginning with simple, direct information, with later texts, levels, or specific challenges within levels relying on previous understanding and practice [7]. As detailed below, the visual language of *Mario* guides the player from the first encounter, *Super Mario Brothers* (*SMB1*) level 1–1, and while subsequent releases in the franchise have modified and added elements, the core actions, choices, and consequences remain relatively unchanged. This empathetic, tiered–difficulty mirrors any number of methods from education, pedagogy, and technical communication: begin with basic, constituent parts, and add complexity and additional elements as facility comes with practice.

However, this basic structure of simple–to–complex, coupled with the increased, and still increasing, proliferation of not only hardware and software, but programming understanding and skill, has disrupted the one–way understanding of designers behind an interface projecting a world for players (like the narrative approach featuring an author and a reader, or consumerist approach of creator and customer) to produce a more dialogic relationship, both within and without of the game–world itself. That is, initially, professional designers carefully crafted these levels. The progression of technology and the availability of tools and knowledge has allowed amateur designers to create levels within the confines of a given game's mechanics to challenge their friends, develop unique puzzles, or randomize the elements of a game to allow players a radically different experience in the game every time they play.

Indeed, anyone with the desire to learn the appropriate assembly language can modify games like *Super Mario World* and create fully–fledged "ROM–hacks"; which James Newman explains as "the direct manipulation of commercially released videogame data so as to alter the original gameplay, graphics and sound or level designs", a practice that is "a clear breach of copyright" [7]. These ROM–hacks afford a sufficiently skilled creator an incredible amount of flexibility with mechanics and game assets; almost anything can be accomplished or communicated given the right assembly code. However, Nintendo's Terms of Use explicitly states "any use of the Services or the Materials other than as specifically authorized herein, without the prior written permission of Nintendo, is strictly prohibited and will terminate the license granted herein. Such unauthorized use may also violate applicable laws, including without limitation copyright and trademark laws and applicable communications regulations and statutes" [8].

While only "specifically authorized" use of Mario is allowed, the popularity and deep love for the game franchise has produced a mass of unofficial and "outlaw" content, which have been as influential, as we will see, as the official content to the overall *Mario* phenomenon. For our purposes, one particularly famous ROM–hack, *Kaizo Mario World*, has produced a whole genre of approaches to level design. Kaizo levels require immense knowledge of game mechanics and near perfect execution to complete. Rather than "gated progression" and moving from simple to complex, according to its creator, Takemoto (about whom little is known) in *Kaizo Mario*, “nothing will spare you, the game will spit on you as if you were nothing but asphalt.” [7].

In contrast to “best practices” of design, Kaizo levels feature “abusive game design” that leverage the designer’s knowledge and experience to anticipate and actively thwart players’ attempts to complete the level, often through deceptive, unfair, or subversive designs [9]. While it begins as an “outlaw,” an "unauthorized” design approach, Kaizo was only made more accessible (and sanctioned) with the 30th anniversary of the *Mario* franchise, and the release of a new product in that Intellectual Property (IP), *Super Mario Maker*.

In 2015, Nintendo released *Super Mario Maker* (or *SMM*) for the Wii U, which simplified the creation of new *Mario* levels, followed by the Nintendo Switch’s *Super Mario Maker 2* in 2019. In these games players can either “Play” levels created by Nintendo or others, or “Create” their own, using a predefined palette of sprites, enemies, powerups, and game themes, and share them with other *SMM* players.

To prevent “bad faith” design, all creators must complete their level before they can share with others [7, 6]. This gate–keeping mechanics of “play first” and the limited tools of *SMM* have produced a wide range of entirely divergent game designs, and it did not take long for Kaizo levels to be created and shared as well, incredibly challenging levels with creative setups using the limited palette of tools available in *SMM*.

Even though the intent is to prevent players from completing these levels, or to make it as difficult as possible, there is still a need for these creators (also players of these games) to meet the same challenge professional level designers do for games developed by game studios: how to teach the player what they need to do. Put differently, it’s a challenge of how the designer–players can help players navigate the Flow Channel, or the balance of challenge in the level with the skill of the player; if it’s “too hard” players can get anxiety and quit; if it’s “too easy” players become bored, and also quit [6].

In ROM–hacks, players can insert arbitrary text and assets, allowing near limitless ways to communicate intent and requirements; however, this flexibility is not afforded to level creators in the *SMM* games. Unlike the broader freedom of ROM–hacks, where instructions can be snuck into the background, the sanctioned tools of *SMM* doesn’t provide enough space to convey the necessary information. As others, including Lefebvre, Johnson, and Newman, have noted, this limitation on the available means to create realities within the game levels and communicate to players means that the “far from allowing for endless possibilities, [the creation tool] actually restricts creative possibilities” [6]. But the restrictions are not simply limited by the available tools within *SMM*, but through the additional protocol of Nintendo’s ownership and curation of *SMM* levels in their hosted “Course World,” which prohibits much direct, didactic communication with players.

Instead of the direct communication afforded by ROM–hacks, in *SMM* the player–makers developed a simple, semi context–aware visual language using the simple primitives provided by the game itself “to hint to players what they should do,” such as “when to spin–jump or where to go.” [10]. Therefore, the language of indicators fills in an important role as “metadiscourse,” combining a series of game elements to act as “commentary,” to direct players to accomplish the goals of their levels [11]. This language has no glossary or dictionary, or even perhaps a written description; it instead relies on the skill of the player to understand the actions available to them to determine what should be done next.

Over time the language has even evolved to account for updates to the game that have added new abilities and graphical elements. This evolution does not seem to be communicated explicitly between creators; instead a creator sees and understands the use of an indicator in a level, and then emulates it in their own; either copying the strategy with fidelity or modifying it for their own unique approach. In this way, the *SMM* Indicators language develops similarly to oral tradition; players “hear” a story (see the use of an indicator) and incorporate it into their own levels, as an allusion, homage, or even outright theft! The understanding of these visual lexemes is left to the level “reader”, whose requisite experience and knowledge is vital to reading and udnerstanding the message and successfully completing the challenge of the level if they have the skills and timing to do so.

This language has essentially been crowd–sourced, creating what Ferdinand de Saussure calls “a community of speakers” outside of which the language “itself” does not exist, and mirroring the change and diversification of dialects in natural language as well [12]. For example, the differences between Spanish and Portuguese develop from differing results of “imperfect copy” of the source–language, Latin. Furthermore, Spanish or Portuguese, and indeed any ‘living’ language, “never exists apart from the social fact”, while the source–language, Latin, is considered “dead” as its power as a social fact has been restricted to scientific terminology and Catholic liturgy [12].

The *SMM* community is comprised of individuals with common interest and immense expertise, so the comparison to scientific and theological jargon is quite apt. However, as this language of indicators is shared globally, with multiple millions of users and even more levels, this “specialized” discourse that functions within a community (like Post–Rome Latin) also changes because of the speed and intensity, wide–base of users, who are both players and creators, readers and authors, and (sans compensation), consumers and designers. Indeed, the impact on language change in the wake of global internet technologies is a potent site of much–needed research beyond the scope of this paper.

Even still, the specific language development and the specific dialogic relationship among player–creators of *SMM* provides insight into how a visual language develops, proliferates, and even changes. The intense practice being a player–maker leads to a common understanding, and indeed at times to setups in levels that are so common they no longer require indicators to be understood, but misunderstanding is still possible, and in the case of "troll" levels, indicators are used to actively deceive players and subvert their expectations.

In this paper, we explore this language of indicators in *Super Mario Maker* community, which illustrates and cultivates the "dialogic relationship" between game–makers and game–players who are the same people; as a microcosm of the effects increased access to computer technology, examining how visual language works among a community of player–makers or user–designers, which is likely to become increasingly common and therefore worth exploring ahead of time.

While we draw on the ideas of "abusive game design", "productive play", and the normative constrains in *SMM*, we focus specifically on how these indicators are functioning as language and visual language, as other analyses of *SMM* have mentioned, but not explored this dimension of the game player–designer community. Indeed, much of the work referenced thus far emphasizes the work done to make a level "challenging" or "seemingly impossible," but alongside that work, and as an important supplement to it, we should look more closely at the ways in which designer–players tell player–designers how to overcome the challenge, even if the player themselves might lack the skills to do so. The indicators are instructions on how to open the door, but it's up to players to walk through it. If they can. If they dare.

To accomplish this task, we will provide an introduction to the relevant limitations in the level creator in *Super Mario Maker 2*. Next, we provide a rough survey of the main styles of *SMM* levels, with a more in–depth discussion of Kaizo and Troll levels, in which the use of indicators is quite prominent. After analyzing some individual elements of the language of indicators, this paper will proceed into a few full screen examples demonstrating how the language can be read in the context of a level. Definitions of relevant player abilities will be provided as necessary. Finally, we will present an analysis of some of the lessons visual language developers can take from such a language that has naturally evolved in an environment of mushroom people and sentient dinosaurs.

2. Related Work and Background

2.1. Visual Language and Level Design in Mario Games

In an interview [13], Shigeru Miyamoto discusses the design of the classic Level 1–1 of the original *Super Mario Brothers*. In this interview, he discusses not only how the level was designed last, but also how it trains the player to play the game through a process of rewards and punishments. On the first screen alone, if the player just walks (or doesn't move at all) the Goomba plodding across the screen will cause the player's death. If the player moves, avoids the Goomba, and jumps into the classic question mark block, they are rewarded with a coin. In this way, the player learns that question mark blocks are positive, and that they must jump on certain enemies to defeat them. However deep pits should be avoided, as should being touched by enemies as these will lead to Mario's demise (or at least a loss of a powerup).

The careful training of players over time into the "official" Mario game–level language is a foundational part of the language of indicators that has evolved over time. By putting the player in situations where they can escape danger or be rewarded by experimenting, the design team trains the player to perform certain actions simply by introducing basic elements that will be multiplied and complicated as the player progresses. This design exemplifies the "user–centered" approach mentioned previously as "good design" of game levels.

The following assumptions will be made about the language that *Mario* level designers have trained players to understand since *SMB1*:

- Coins are rewards, and they should be collected (motivating the player to move Mario to collect them)
- Question mark blocks, disguised question mark blocks, and turn blocks can be struck from below by the player (or from the side using a thrown item which is introduced), which often rewards the player with a power–up or some other form of level progression
- Enemies can be safely jumped on unless they have some form of protection (like spikes or fire); later games gave Mario the ability to safely land on these enemies using a spin jump move.

The core mechanics of the game are exposed to the player through low–stakes experimentation in the first few moments of the first level, not explicitly through text. This is truly a marvel of game design principles, and it exemplifies Nintendo's " 'user–centered' approach to game and level design that places considerable emphasis on engineering particular kinds of experiences for the player...gradually introducing and ramping up challenges" [7]. The mechanics and visual elements from *Mario* games are the foundation for level creation in the *Super Mario Maker* games, released in concordance with the 30th Anniversary of the original *Mario*.

In like fashion, when players first open *SMM*, they are given a choice of "Create" or "Play": either delve into the Maker side of the game, or play through the results of others' efforts, whether official content from Nintendo or any of the over 26 million player–created levels (as of 2021) [14]. While many of the levels created explicit reject "good design", Nintendo still uses the informative–design that has been characteristic of *Mario* games since Level 1–1. As LeFebvre explains "when a player plays *Super Mario Maker* for the first time, she starts in a level reminiscent of the first *Super Mario Brothers* (Nintendo 1985)", which plays directly into the established expectations of *Mario* games [6].

Although, at the end of the level, there is "a bottomless pit that is too wide for Mario to traverse", and when the player inevitably fails to make the jump, rather than return to the start a message appears "Whoa! Looks like someone left this course unfinished...It's up to you to complete it", after which the player is provided a basic Maker toolset to complete the level themselves [6]. That is, in a game franchise built from its earliest manifestation of providing in–game tutorials using visual indicators to teach the player, *SMM* provides the same experience in the emphasized element of this two–sided program, "making of *Mario* levels by the player" [6]. They also do so through direct commentary to the player.

However, even though making is emphasized within the first level that can be played, Nintendo also requires all levels to be completed by their creators before they can be shared [6, 7, 10]. In the same way that Create mode is emphasized within Play mode, so too is Play mode emphasized within Create mode. Furthermore, player–makers can switch between modes at will, showing how this game–development game disrupts the common monologic, monodirectional practice of developing a game using special skill and insight for publication and consumption by players who may or may not share some of that skill, but certainly don't rely on it to play the game.

2.2. Limitations in Mario Maker 2

The way *Mario Maker 2* conveys information has two major limitations. First, each screen of the level is a grid that is 24 "blocks" wide by 13 or 14 blocks tall. This space limits what knowledge can be conveyed through simple "pixel art". Additionally, the only free–form text that can be distributed with levels comes from the level description, which has a limited character count. Players can leave small comments in the level, but they may not always be trustworthy or helpful (as we will see with "troll" levels below).

Additionally, because the interface for both play and design are the same, and all items and design tools embody "the aesthetic of Mario's worlds," Newman argues that "*SMM* gamifies game–design", rendering a space to not exactly Make Mario, but rather a space to Play at Making Mario, while the real tools used by the original designers are left to automation or out of the player–designer's control [7]. That is, *SMM* does not provide the real tools and approaches of the Nintendo designers, any more than Mario provides real training in plumbing, but rather a sand–box and toy–box producing something slightly different.

Furthermore, Nintendo is a corporation interested in creating profits, for which games are products, the results of budgets and timelines. Indeed, as LeFebvre argues "the *Mario* franchise does not depend as much on the development of rich narratives within a complex fictional universe as it does on the diversification of cultural products and experiences for a commercial purpose." [6]. Certainly, a princess being in another castle and the increased amount and intensity of antagonists do not attempt "world–building," and Mario's own complete lack of character development over almost 40 years push it away from being a strongly narratological game; the proliferation of Mario spinoff games, like *Mario Kart*, *Mario Paint*, and *Super Mario Maker (1&2)*, further support LeFebvre's position, in addition to the merchandising that have placed Mario alongside any number of other immediately–recognizable cultural figures of late 20th and early 21st century culture writ large. In addition to being something a user plays, Mario is also something a customer buys.

As a product, the "good design" of progressing simple–to–complex, easy–to–difficult, managing the challenge to offset both anxiety and boredom, is easy to understand. If players (users/customers) find something accessible and usable, they will buy it, and keep buying. But as a product, an Intellectual Property (IP) still owned by Nintendo, the "freedom" of *SMM* is constrained by the limits imposed by the company itself. While anyone can make and upload levels to "Course World" to be played by others, these creations exist only within the legal bounds outlined by Nintendo.

Levels can be deleted for "Inappropriate Content," as defined in the Nintendo Network Code of Conduct, which is vital for a "family–friendly" franchise like *Mario* [7]. Additionally, and interestingly for us, but not something to fully explore here, levels can be deleted for including "bugs... because letting these levels remain in Course World [the shared collection of *SMM* levels] can lead to negative outcomes for many players such as players experiencing levels in unfair ways that the original course creator did not intend, or re–writing 'World Record' times" [7].

Considering the essential "unfair" nature of "abusive game design" in general and the "the creative utilization or even abuse of object/item behaviours and interactions," in Kaizo design in particular, makes this process "especially problematic" because it isn't always clear "what constitutes a 'bug' at the point of level creation and uploading. That the decision to designate as a bug what might have previously been apprehended and utilized as a creatively exploitable behaviour is, demonstrably, one taken by Nintendo and leaves levels created and uploaded in the liminal zone before the recoding (and potential removal/recalibration of the 'offending' behaviour) vulnerable to deletion" [7].

Even still, Kaizo level designs remain the most popular and pervasive genre of *SMM* levels, which exemplifies the creative work of player–designers, and these player–designers develop their own aesthetic style and followings themselves, which can lead to other connections and opportunities [7].

Furthermore, "after a fixed period of time, courses with low stars/plays" are deleted automatically, making successful completion, shown as a percentage, and subjective satisfaction of the players, indicated by stars, as well as motivation to share or play the level itself, a vital part of levels' survival in the massive Course World [7]. However, levels are also deleted for "requesting stars from other users," which makes the direct communication with users not impossible, but risky. Therefore, to maintain the existence of levels, even (and especially) complex and difficult ones, designers have a strong motivation to not only challenge players, but to provide insight into how to complete these challenges.

Even with these limited affordances, and sanctioned tools that differ from the original designers', players have made ample use of available resources, and the flexibility surrounding them in the *SMM* toolbox, to communicate directly with players to help them complete challenging levels.

2.3. Kinds of *Mario Maker* Levels

Creators can create any level they can imagine if it is completable and complies with the terms and conditions Nintendo places on the levels it hosts. This wide range of levels also occupy different places within the ludic–narrative spectrum of video games. This range has produced several distinct types, or genres, of level design, each with a different design approach, effect, aesthetic, and particular audience. Some approaches are standard levels that would be appropriate for inclusion in a commercial Mario game. These levels can be understood as the most literal and direct result of a game called *Super Mario Maker*.

Other levels are "AutoMario", in which the player does nothing or simply moves forward, the level having been meticulously constructed to reward the player's inaction with a cinematic, dramatic sequence of Mario braving hazards and only–just–escaping certain doom before arriving safely at the end. Still others emphasize a narratological approach within a level, complete with dialogue, while others feature challenging puzzles, a ludic approach, that work within the constraints of *SMM* and the game's longstanding influence.

However, the most popular style of level, as discussed previously, is Kaizo, which uses the limited toolset within *SMM* to emulate the difficult and complex designs of the ROM–hack original. First appearing in a video in 2007, *Kaizo Mario World*, or "*Jisaku no Kaizō Mario (Sūpā Mario Wārudo) o Yūjin ni Purei Saseru*" which translates as "Making my friend play through my Mario hack (Super Mario World)", features the extreme opposite of Nintendo's game design best practices. Rather than Nintendo's advice, "you want the player to feel like they're about to get crushed," producing enough challenge to motivate play without quitting, in *Kaizo* "the player not only feels like they are about to get crushed, they get crushed. Many times over." [7]. Drawing from the ROM–hack approach, Kaizo levels push the limits of the mechanics afforded to the creators and the abilities of a player to jump, aim, and time everything just right. These levels often require precise tricks and near perfect execution to complete, and the payoff in these levels is the satisfaction that the player has obtained a level of skill well beyond that of a causal player. Indeed, the prevalence of Kaizo levels, as well as play–through videos on Youtube and Twitch, has become the most visible examples of *SMM*, and bringing back those hardcore gamer fans of the franchise who might otherwise pursue ROM–hacks in search of a challenge.

Additionally, this purposefully–nigh–impossible design in Kaizo is accompanied by a "markedly more dialogic relationship between player and designer," which Newman cites as characteristic of "abusive game design" coined by Wilson and Sicart [9]. While so–called "good design" causes the designer's presence to "recede into the background" becoming "anonymous" (in the same way "good" language use is "clear", emphasizing the erasure of the words themselves), abusive game design in Kaizo creates a "clear exchange existing between designer and player," with the level being "the site of their contest" as "one between two (or more) human participants [rather] that one between a player and a system" [7]. That is, a Kaizo level is more about the player and the designer than the level itself, foregrounding the communicative medium of the video game as one between people; the game is an instrument to build relationships.

Furthermore, as levels are shared across the Course World, they are "assiduously attributed to their nicknames" and named by creators, and "once one alights upon a designer whose levels one enjoys, they can be followed, ensuring no new upload is missed and cementing the relationship yet further in much the manner of social media" [7]. After all, the original *Kaizo* video is making a friend play a challenging game.

However, not all purposefully–nigh–impossible designers aim to reward skill and timing alone, but some levels require in–depth thinking and suspicious play. In Kaizo levels, the challenges are extreme and arduous, but essentially honest; in contrast, Troll levels are extreme, arduous, and frustrating, as they are they are built on deception. A relationship is built here too, but through a practical joke. Troll Levels are similar to Kaizo levels, but they are designed to subvert the expectations of the player, often leading to an excessive number of character deaths. Often, troll levels will evoke schadenfreude in those watching the player, and they are entertaining in the creative ways the player can be killed more so than in the successful completion of the level.

These levels rely on the learned expectations of expert players, second– and third–level thinking, to encourage them to "troll themselves," as described by Johnson. Drawing from studies of poker, Johnson explains first level thinking as "simply looking at the level as presented, while second level thinking means considering that the designer was thinking, and third level means considering what the design thought the player would be thinking at a given movement and what decision they (the player) might be poised to make" [10]. These levels will often provide the player with one or more "obvious" paths to take, then punish the player when clever mechanisms cause a new path to appear after the player has committed to the wrong choice.

These levels can be often removed because of the presence of glitches or unintended mechanics, but they have a cult following where people will upload a level long enough for it to be downloaded by others before it is removed from Nintendo's servers. Streamers on services like Twitch will hold contests to see who can build creative trolls [15]. Their popularity likely stems from the comedy that comes from creatively subverting expectations and the streamer's reactions to the creative and unexpected deaths. The use of indicators adds to the psychological duress on the player as there is never the certainty that an indicator is lying, telling the truth, or even partially lying.

3. A Language of Indicators

Building on the previous assumptions (and referencing them), this paper will focus on three extensions to the visual language: the P–Block, a C–shaped track, and a curved track. These indicators communicate modifications to position, action, and movement, respectively. All images below were captured from the author's console using the screenshot feature and edited as necessary. Interestingly, it may be almost impossible to track down the first use of these indicators; as discussed above, levels sometimes get removed for a variety of reasons and as such the genesis of some of these indicators and any textual description they might have had may be lost.

While these visual lexemes are only a selection of the various tools available and in use, they provide the most telling examples of the "visual metadiscourse" of indicators at work in *SMM*, which cultivate the "dialogic relationship" between player and designer central to Kaizo design. Drawing on Vande Kopple and others, Eric Kumpf explains that "metadiscourse helps writers arrange content by providing cues and indicators that both help readers proceed through and influence their reception of texts...To omit this metadiscourse would blur the separation of content, making the text less cohesive and less considerate of readers" [11].

While discourse is the communication, exploration, and presentation of the ideas, arguments, and impressions of a text, metadiscourse is writing that helps discourse accomplish its goals. That is, metadiscourse is writing explicitly designed to help the reader read and understand a text, and as Kumpf explains, this is not limited to textual commentary (like "in this paper will first, then, next, later...,") but also visual elements, like abiding by convention, maintaining consistent design, and dividing content into accessible chunks, "may help readers understand and interpret the whole document." [11]. In like fashion, level designers (text writers) use metadiscursive indicators to help level players (text readers) to read the level and understand how to complete it.

Furthermore, like all semiotic units, these indicators can be understood through the structural linguistic frame of Ferdinand de Saussure. While de Saussure's initial discussion is primarily focused on words in phonetic languages, it also applies to the elements of visual languages. Rather than the indicator, or token, or *Sign* being a singular entity, de Saussure identifies two interconnected, yet distinct, elements: the *signifier* and the *signified* [12]. The *signifier* is the word the reader sees or the sound the listener hears. In the context of visual languages, the graphical elements constitute the signifier, that which is seen by the user. The *signified* is the meaning that the listener (or reader in the case of written and visual languages) attributes to the signifier, the idea intended to be thought about, the concept being communicated, or the action intended in response.

The *SMM* Indicators are then Signs, combinations of a graphic element drawn from a collection of possible elements with a corresponding idea or action for the player to actualize in response, and the language a Language, "a system of distinct signs corresponding to distinct ideas." [12]. This distinctness is central to coherence and communication: "M" is a distinct shape for a distinct sound, because it's different from all other shapes/sounds, and by adding other distinct shapes and sounds, a fuller distinct concept can be communicated, "Mario" (with further distinctions needed to specify which Mario is intended, whether real chef or fictional plumber). However, the connection between the signifier and signified is not natural, but the result of conventional use and rhetorical repetition.

As a First Principle of semiotics, the study of signs (of which linguistics is one, yet paradigmatic, example), de Saussure argues that Signs, combinations of Signified and Signifier, are arbitrary. That is, a signified idea is "not linked by any inner relationship to succession of sounds...which serve as its signifier," and "it could be represented equally by just any other sequence [, which] is proved by differences among language and by the very existence of different languages" [12]. As a Second Principle, emphasizing phonetic, spoken language as the primary "object" of study, is that the signifier, the sound sequence, is linear, which is "obvious," "too simple," and therefore assumed, but yet "fundamental, and its consequences incalculable." [12]. Indeed, as a corollary to Principle II, de Saussure contrasts the "auditory signifiers [that] have at their command only the dimension of time," with "visual signifiers...which can offer simultaneous groupings in several dimensions" [12]. From these two principles, de Saussure developed a massively influential theory that shaped inquiry in the humanities and social science quite broadly, not limited to language study.

It is interesting to draw comparisons between de Saussure's structural linguistic frame and the established frameworks for visual languages, such as those by Green [16], Moody [17], and others, because the structural approach not only precedes any development of (and analysis of) programming languages, but it would also therefore inform the understanding of "language" as developed within programming, visual language included. The nature of this work also relates to the field of Computational Semiotics, but is outside the scope of this paper. For more on this field, see [18].

Chang et al. discuss the use of formal language theory in relation to visual languages, but call for an interdisciplinary approach to extend the field [19]. The formalisms specified by formal visual language theory apply when working to intentionally design a visual language, but there are aspects that fall apart when dealing with a language changing in real time. De Saussure explicity addresses these different approaches; there is a formalistic, synchronic study of language, drawing on the method of comparative philologists, whose work begat proper linguistics for de Saussure, and in contrast, a diachronic study of language examines how language use changes in time [12]. While limited in his time by technology, de Saussure's emphasis on the importance of structure, rather than use itself as in speech, is understandable; in the *SMM* Indicator language, we can see, though only in part, the development, proliferation, and change of a distinct, visual language with a massive community of speakers.

This is not to say that the standing theories of visual languages are at odds with structural linguistics, though we may need to interpret elements in both fields with some flexibility. For example, Green's Cognitive Dimensions [16] address the closeness of mapping of a notation to the problem world, which would at first seem at odds with de Saussure's idea that the signifier is arbitrary. In a visual language, however, closely mapping to the problem space means the signifier is chosen specifically to communicate the signified. As a "metadiscourse" the closeness of a signifier to the signified helps assist the user by relying on already existing connections outside of the text, and because the goal is not to vocalize the signifier, we are not limited to phonetic signifiers.

Furthermore, de Saussure includes "modes of expression based on completely natural signs [closely mapped], such as pantomime" among the science of signs, because "arbitrary" does not mean totally random, but only that a specific signifier "actually has no natural connection with the signified" [12]. Indeed, "every means of expression used in society is based, in principle, on collective behavior or–what amounts to the same thing–on convention" [12]. Calling a herbivorous quadruped with mane and hooves a "horse" isn't more true or accurate than calling it a "*caballo*" in Spanish, or the Portuguese "*cavalo*" or "*égua*" (depending on gender), or the Latin *equus*; put differently, a horse isn't any more "horsey" than it is equine(although Latinate roots in English are used widely within specialized, scholarly discourses to afford so–called precision), but the effect and effectiveness of those Signs will depend on the context, convention, and situation.

Moody's Principle of Semiotic Clarity [17] particularly falls directly in line with de Saussure's concept of a sign being the union of the signifier and signified and the immutability of the signifier within a given community of speakers. Moody does specify that there should be a one–to–one relationship between signifier and signified in visual notations, and while de Saussure is more relaxed in the definition, due to the messiness of natural languages changing over time, he is explicit that distinction, one shape to one sound to one idea, within a community of speakers is constitutive of the language itself. Indeed, a sign "is never wholly arbitrary [totally random]; it is not empty, for there is the rudiment of a natural bond between the signifier and signified. The symbol of justice, a pair of scares, could not be replaced just by any other symbol, such as a chariot" [12].

In this sense this principle of the Physics of Notations is not a departure from the principles of linguistics, but a subset, and while Sassure grapples with the complexity of language being conventional, but not random, but not natural either, the Physics of Notation must work toward more clarity. Put differently, programming languages work to create an ideal language with much less ambiguity and misunderstanding than natural language; code works or it doesn't.

However, as we've seen with ROM–hacks, Kaizo level–design, and trolling, and as de Saussure states that convention is not total control, which "even holds true for artificial languages. Whoever creates a language controls it only so long as it is not in circulation; from the moment when it fulfills its mission and becomes the property of everyone, control is lost" [12]. That is, language, a series of connections of signifier and signified conventionally constructed and used by a community of speakers, changes and persists, and once out into that community, and beyond, the creativity of those speakers moves it well beyond the control of the originators.

Unless, of course, they violate Terms of Service, IP, or Copyright, laws and rules that help originators, like Nintendo, maintain control. However, the challenge of communication is embodied in fictional voices, like Yamamura, a pigeon that inexplicable understands the nuances of level design in *SMM*, even if the effect is an erasure of the designer, fading back into the system itself, which is the main way *SMM* communicates with player–users.

But whether they're ROM–hackers illegally ripping and remixing source code, *SMM* designer–players creating complex level–texts out of a limited set of items, Kaizo–designers eschewing 'good design' in favor of 'abuse' with indicators to help understanding, or Trolls using the same indicators to lie and deceive, and 'abuse' their players, even with legal weight and force, there is movement, change, and evolution.

However, it's important to note that difficulty and deception are not necessarily correlated to the popularity and fun of "abusive" levels; Kaizo and Troll levels use indicators differently, and the quality of their use shows the quality of the level. That is, both Kaizo and Troll levels are manifestations of "abusive game design" and examples of "masocore" gaming, "the portmanteau...combining 'masochism' and 'hardcore"', which emphasizes repeated attempts and failures on "punishingly difficult" challenges [10, 7]. The "fun" of these designs might be called "suffering" by more casual gamers, and coupled with the powerful schadenfreude provided by streaming play–throughs, "abusive" design is a powerful force. However, in a set of game designs where difficulty and suffering is the point, there are still distinctions between "good" and "bad" designs, and the use of indicators is a key factor in those distinctions.

Kaizo levels follow the tradition of the first *Kaizo Mario* game, which was intended to both challenge and frustrate the creator's friend. These levels have varying use of indicators to show the intended actions, but they still rely on the player's skill to complete the level. The challenge in these levels is not only in determining how to reach the goal, but also the execution of the level itself.

A "Good" Kaizo levels "presents a player with a unique set of precise gameplay challenges which ideally look compelling or beautiful to perform," while a "Bad" Kaizo level is marked by "its lack of creativity, its pure repetition, its almost complete lack of aesthetic detail, and its absence of pleasing sequences," which Johnson sees as evidence for "new *surprises* being indicative of good abusive design" [10]. Creativity, beauty, nuance, and craft, as well as being playable with hours of practice, or not, are all central to Kaizo design, and indicators form a central part of that; Kaizo indicators show the player the door, even if they might not be able to see how to walk through it. It's the joy of a challenge overcome.

Similarly, Troll levels will often (but not always) lie with indicators, following the tradition of subverting a player's expectations. Johnson explains that "a good troll level is one which toys with the player's expectations, gets into the player's head, encourages the player to get into the designer's head, and perhaps most importantly...provokes entertainment (falling into a trap) just as much as it provokes frustration (at falling into a trap)" [10]. Some levels even work to subvert the expectations of expert players, providing an obvious path to success within an otherwise 'normal' Troll level, leading an expert *SMM* to "troll themselves."

In contrast to the nuanced, psychological approach of "good" Troll designers, a bad Troll level "is one where your required choices are arbitrary (such as just choosing a door at random), or the level is packed with enemies to the point where nothing except purely chance–based trial and error will get you through."[10]. Creativity, beauty, nuance, and craft, as well as being playable with hours of practice, or not, are all central to Troll design as well, and indicators form a central part of that, even if they are not to be trusted as in Kaizo.

That is, the "thin line" between good and bad levels, for both Kaizo and Troll levels, seems to be their use of Indicators, reflective and purposeful construction of metadiscourse (even if it isn't always 'honest', but communication between humans is rarely as full, honest, and complete as we might imagine or hope). This metadiscourse and the "dialogic" relationship that it creates is foundational in "abusive" game design, in which the player experiences the designer's presence and persona, through both aesthetic and direct communication with indicators.

These indicators are motivated by Nintendo's rewards for plays and stars, and the relationship between designers and player, as well as the prohibition of explicitly soliciting stars. In balancing anxiety and boredom, player–designers in Super Mario Maker use indicators as metadiscourse in their levels to make them "considerate texts", to be read and understood by their readers, even, and especially, when consideration for the player is couched within abusive game design. It's not a matter of failure, in Kaizo or Troll, but that the player understands how and why they fail, and they have fun doing it.

What follows is a subset of the language of indicators. These are specific examples that help inform the "reading" of the level presented in Section 4 or are particularly interesting in how they signify meaning.

3.1. The P–Block: "Aim Here"

In *Super Mario Maker 1* and earlier updates for *Super Mario Maker 2*, creators were limited to using the coin to indicate to players where they should attempt to send their character. Due to the limited palette, the coin was a multi–purpose reward indicator, promising progression if the player performs the correct action at that location. Level creators were afforded a new option with a later update to the game: the P–Block (Figure 1), which occupies a single grid space on the screen.

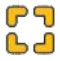
Figure 1: The P–Block

Mechanically, when this block is in the state shown in Figure 1 it behaves like a background object the character can pass through. If the character hits a P–Switch, the block becomes a solid object, which can serve as a platform or wall. The background state of these blocks makes it useful as an indicator, but particularly clever creators can use it as an indicator in the background state and as a platform or wall in its activated state. The P–Block is especially useful when the item or enemy the player will be landing on is not yet apparent but will be by the time Mario reaches that location, assuming all preconditions have been met.

This indicator then acts as a general "aim here" message to the player, and is used in many of the situations the coin was in earlier releases of the game. This change allows the coin to be used as a different kind of indicator, which illustrates de Saussure's point that "everything that changes in the [language] system is internal" [12]. While the new signifier P–Block takes up some of the signified action of the Coin, this change in aligning a new signifier with part of the signified of another signifier is an internal change: a change in the connection of a signifier to a signified. The Coin no longer means what it did, signifies the same signified, as it did before the introduction of the P–Block. Furthermore, and interestingly, this indicator's meaning can change based not only on how it is used in a level, but also within the style used to create the level.

Visually, the negative space in this indicator in its inactive state somewhat resembles a cross–hair or target, which evokes the real world idea of aiming for a particular location (an example of Green's Closeness of Mapping and de Saussure's emphasis on convention). While the color cannot be changed by the level creator, extra meaning can be conveyed through placement or number of indicators (which is related to Green's Secondary Notation and Escape from Formalism). For example placing one P–Block directly above another often indicates that the player should land in that location twice. An additional example is provided in the partial level read in Section 4.

This indicator can be confusing if there are multiple reachable from the player's current location, making Premature Commitment particularly a concern at times, though often after one or two mistakes it becomes more clear as to which location should be the player's next target. As discussed previously, Indicators "communicating effectively" depend on the audience, the purpose, and the context of its use, and players need the skills to read and interpret what the designer had in mind.

3.2. Shaped Tracks: "Throw it this way!"

In all level styles except *SMB1*, the player has the ability to pick up, carry, and throw objects. These sprites can interact with the world, triggering switches, breaking blocks, or just providing an object the player can land on later. Kaizo levels often require objects to be thrown precisely to allow the player to proceed. As quite flexible game–objects, tracks can be placed in relatively arbitrary shapes, but must either be a closed loop or a simple path. Often, a closed square loop in a 3x3 grid is used to indicate to the player that they should wait at a location (or something will appear there). Alternately, Tracks can be placed in the shape of a "Z" to indicate that the player should press the Z button on the controller (this is a fairly explicit use of indicators to communicate intent to the player, as part of the "dialogic" relationship). While arrows pointing in various directions can be placed in levels, creators often rely on a ⊏, ⊐, ⊓, or ⊔ shaped set of tracks (Figure 2).

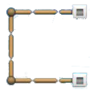
Figure 2: Tracks shaped to indicate the direction to throw a held item

When one side of the closed square loop is omitted, it tells the player to throw or drop the item they are currently holding in the direction of the missing edge when they are inside the square. Usually if the player releases or throws the correct held object in the direction indicated by the opening while Mario is in the outlined box the item will end up exactly where it needs to be for the next required steps in the level to barely work, only just so.

This indicator is interesting with respect to the Cognitive Dimensions of Notation. The size of the symbol is larger than the corresponding arrow (3x3 instead of roughly 1x3), but provides additional context of the player's location when the object should be thrown, as a metadiscourse commentary [11]. Choosing to trade terseness for additional information may depend on the available space. Arrows are also frequently used to indicate a direction the player should go if it's not immediately clear; this may be a reason why the shaped track was chosen.

Additionally, this particular indicator has a hidden dependency: the player must be holding an object for it to have meaning, and if there are multiple objects available they need to be holding the correct item. If a player reaches this indicator without an object, it may be a hint that a previous part of the level has been misread or the player did not execute the previous parts of the level as intended (and instructed).

Also, there is nothing obvious about this particular indicator that indicates throwing. The closest it comes is indicating a position and a direction. Closeness of Mapping may be this indicator's weakest Cognitive Dimension, though there is not a particular symbol available that would necessarily directly communicate the idea of "throw" to the player. Rather, this signifier has connected to the signified through conventional use. However, in concert the mechanic of holding and moving items, the conventional assignment of "tracks" with "movement" and the repeated, conventional use of this indicator to communicate the intent takes up the communicative work than a more Closely Mapped indicator would perform.

One Cognitive Dimension this notation rates particularly well at is Consistency, a name shared by one of Kumpf's visual metadiscourse devices [11]. The orientation of the notation (which side is open) indicates the direction the player should send the held object. Once a player has learned that the ⊏ shaped track means "throw the held object right", the convention at play, the player can then infer that ⊔ shape means throw the object up (in game styles that support that), a ⊓ means drop the object, and a ⊐ shaped track indicates that the object should be thrown backwards. Once we learn about a distinct sign, modifications to it, which combine conventional understanding of the significance of the sign itself with other visual language elements, like direction, we can extrapolate one sign into many, each with a distinct signifying form and a signified action/concept.

3.3. Curved Track: "Twirl Jump Here"

Super Mario Maker 2 allows an additional trick with Tracks: a Track placed diagonally can curve instead of going in a straight line (as in Figure 3). Unlike the track configuration in Figure 2 that encodes its meaning in the empty space contained within the track and its missing side, the path this indicator expresses its meaning in its shape.

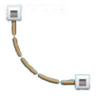
Figure 3: Tracks curved to indicate when to twirl

In the *New Super Mario Bros. U.* and *Super Mario 3D World* level styles the player can perform a maneuver called an air twirl. This technique is performed during a normal jump. Players make Mario spin once in midair which briefly stalls the character's downward momentum. This effectively extends the reach of the jump slightly as in Figure 4.

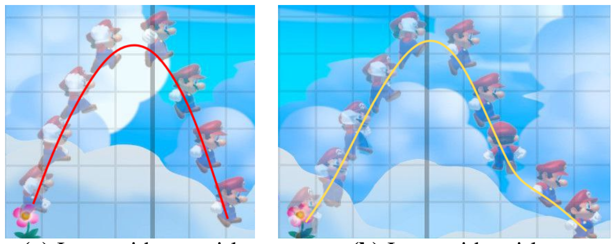
(a) Jump without twirl **(b)** Jump with twirl

Figure 4: A comparison of Mario's jumps with and without an air twirl

Note the curve in the path at the end of the jump in Figure 4b.Mario's path is almost exactly the shape of the curved Track, an excellent example of Closeness of Mapping. Treating the grid as a standard Euclidean graph, Mario begins the air twirl at $(6, 2)$ which stalls his downward momentum as he travels to $(7, 1)$; those are exactly the grid spaces the track would occupy; beginning an air twirl at the top square of the track will cause Mario to follow the path indicated by

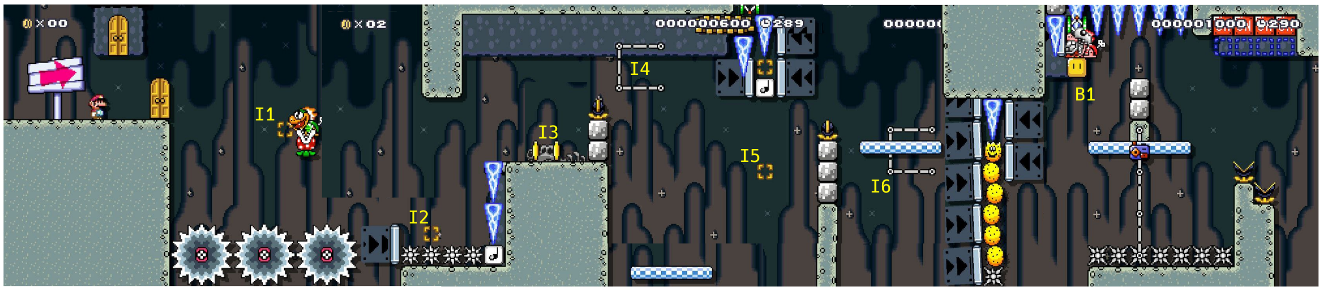

Figure 5: A partial level with indicators. Footage of a player reading and completing this level can be found at [20]

the track very closely. Here the "natural sign" of movement in space is illustrated by the shape of the indicator, but the conventional understanding of what moves are available in what styles of *Mario* is also necessary to understand the message. A player who understands the mechanics afforded by the level style can read this indicator and map that to the necessary controls to make the character follow the path shown by the notation. Like other Indicators, within and without Mario, it is arbitrary, but not empty.

4. Reading the Level: An Example

Figure 5 contains an annotated representation of the first few screens of the level Mechanical Manacles created by Donkeymint (Level Code 8QH–JRX–GLG). This image was created by splicing live game play images captured directly on the Nintendo Switch, and it is annotated to explain how a player who knows the mechanics of both the game and level style can complete the level with enough skill or practice. The difficulty of playing these levels, as well as access to Nintendo products, can make examining these levels directly rather challenging itself. Luckily, the massive popularity of *SMM* and the proliferation of level–play videos provide ample secondary sources to examine.

The player begins on the left side of the screen. The actions that must be taken do have a strong ordering, and it is not immediately obvious how to get from the starting location to the first safe platform (the coins at indicator I3). Because the timing of the actions is also critical, skilled level creators will sometimes add "reset doors" as seen at the beginning of the level to give players time to analyze the situation or retry a particularly difficult trick, which helps avoid premature commitment. The indicators themselves do not necessarily have a way to indicate order of actions, often the only way to determine what action to take next is to look at what the only reachable indicator is. While the indicators themselves do not necessarily follow de Saussure's Principle II of linearity, their sequence does, as the level itself exists in time for the player.

While there is no immediately clear path forward, the indicator I1 tells the player that they should land there. Because this level is in the style of *Super Mario World* there are two choices of how to get there: a regular jump or a spin jump; one of these jumps will lead to success while the other will cause the player to fall into the Piranha Plant (the bottom enemy in the stack) or the saw blades below where they will die. Since there are only two options and this is the first jump in the level, failure is not too punishing; however a player who understands the mechanics of the game can glean more information from the indicator.

I1 is placed to the left of the top enemy, called a Mechakoopa, and if the player was simply using it as a way to bounce higher, then the player would land directly on top of the enemy. However, here we see another convention at work: not every enemy reacts to jumps in the same way; the player needs either learn through experience or research how enemies interact with different player actions. The placement of I1 indicates some directionality in the desired end result. With a regular jump, the Mechakoopa would collapse and fall straight down (or be knocked slightly to the side), but a spin jump will send this particular enemy in a direction opposite to the side they were hit from (if they were hit from the left, they would move right, and vice versa) [21].

By using a spin jump on the Mechakoopa at I1, the enemy will be stunned and sent to the right, landing on or near the indicator at I2 (depending on the exact timing of the jump). Because the character continues spinning, they can land on the stunned Mechakoopa and safely make it to the coins at I3. At this point, the next indicator (I4) tells the player that they should throw something to the right, but currently has nothing in hand to throw. Depending on the exact timing of these first two jumps, one of two things will happen. If the player landed again on the left side of the stunned Mechakoopa it will be knocked to the right where it will land on the note block and bounced up to the top of the icicles (directly next to the player). But if the player doesn't knock the enemy to the right, it will eventually recover and begin walking around, turn around at the one–way gate to the left of I2, and then walk to the note block and bounce up to the player waiting at I3. At this point the Mechakoopa can be picked up (or re–stunned and then picked up), giving the player something to throw.

After jumping and throwing the enemy to the right at I4, there is only one safe place to land: the checkered platform below the indicator. However, this platform will start to fall quickly, so the player will need to quickly decide what to do. The next indicator (I5) tells the player to jump there, but at this point there is nothing to land on. However, when the Mechakoopa is thrown from I4, it lands on top of the note block to its right, which spawns a flying enemy directly at I5. The player can then bounce off of this enemy and land on the checkered platform that is in the middle of indicator I6.

Once again, the player is told to throw an object from a location but isn't holding anything. The mechanism above the note block above I5 causes the Mechakoopa (which was thrown at I4) to bounce up and then over, landing on the checkered platform. This enemy should then be thrown at I6; there are P–Block indicators behind the tall yellow enemy (a Pokey) that can be seen as the enemy moves. The Mechakoopa destroys the Pokey and falls to land on the spike at the bottom of the screen. A spin jump will allow the player to jump off of the stunned Mechakoopa and land on the right most checkered platform, which then begins to move up and down.

At this point the creator forgoes the use of custom indicators and relies on the visual language established over time by the developers of the *Super Mario Brothers* series. After landing on the right most platform (which begins to move), the player has exactly one choice left: hit the turn block B1. This triggers the mechanism above the turn block, which sends another stunned Mechakoopa to land on the right side of the checkered platform. At this point the player again has exactly one choice that does not result in death: pick up the stunned Mechakoopa and throw it up into the On/Off blocks which are hidden by the timer in the upper right corner of Figure 5. Hitting these blocks will cause blue blocks to become solid and red blocks to become background objects, allowing the player to progress further in the level.

An important consideration for this level and the spliced image is illustrated in Figure 5. The images were captured by a player who has not devoted hours of practice to develop the skill that professionals will hone over time. In fact, the amount of skill required to beat the level is independent of the knowledge and ability to translate the language; despite being a mediocre (at best) player, one author was able to progress this far in the level through repetition and practice; the intended action was always clear, even if the muscle memory and skill was not developed enough to allow successful execution of every required move. The other author, to the best of his recollection, has never successfully completed a single *Mario* level alone.

5. Discussion and Conclusion

Language designers can take multiple lessons, or reminders perhaps, from investigating what we've shown here. Visual languages are often used to lower the barrier of entry for individuals completing a new task, as metadiscourse to make a text more considerate of its users, as Kumpf puts it. But experts create mechanisms for communicating succinctly that require a learning curve. For example, experts can use shorthand, a glyph–based method of shortening written communication, to increase the speed with which information can be recorded. Quickly decoding shorthand requires practice and knowledge of which system encoded the information. While useful, such methods are not explicitly designed for simplicity or guiding novice users through completing a task.

This language of indicators serves much the same purpose as shorthand: meaning must be conveyed using limited resources in a constrained space. A visual language is used because unrestricted words and lettering are mostly unavailable to level creators. By removing the expectation that the language be immediately accessible to novice players and their knowledge of the game mechanics, the language can be concise and still convey varied meanings with a single indicator based on the context in which it is used.

This is not to say that accessibility is not important; in fact visual languages should continue to be an important part of aiding in accessible design goals, regardless of whether we're working on digital interfaces, analog written documents, or any of the wide range of texts and mediums available to contemporary designers and users. The role of visual languages in wider accessibility is beyond the scope of this paper.

Even still this language of indicators enables communication within an audience that shares common knowledge and skills, and in the case of *SMM* to reward that knowledge and skill. Understanding the target audience is important when devising any form of communication, as the conventional understandings of one community of speakers are not going to be the same as another. Domain specific visual languages should be designed with a specific audience in mind, and the intended audience needing certain domain specific knowledge to comprehend the visual language should not be a barrier to the design of the language.

The language of indicators is a prime example of "designing" a language with a specific target audience in mind. How often is the user experience and functionality of a deployed system designed and developed seemingly without consideration for the intended users? Updates resulting in removal of used features or strange UX choices often result in members of the user base reaching out to each other and the developers of the system as in [22]. To quote one reply from that particular thread: "Any time we need to tell instructors 'you have to redesign your course / assessment to fit into the software constraints' you know there is a poor software design." Any visual language must be designed and implemented with the target audience in mind, and while there must be common understanding for any communication to happen, we cannot assume that the audience will be of the "same mind" as the author. Indeed, the kinds of "level thinking" employed in Troll design are essential here; it's not enough to see what a text looks like to us, as makers, but also what we think it will look like to users, readers, and players.

However the most intriguing aspect of this language is that it was designed by players for players. Unlike the Hmong Script or the Sequoyah Syllabary, with a single creator endeavoring to preserve a language from annihilation, many people have built the language over the life cycle of two different *Mario Maker* games. This language has evolved as updates have changed elements used to indicate meaning, external changes in the language that lead to internal changes.

But tracing the origins of specific indicators, like "throw" are impossible because the online services for the original game have been shut down, and levels can (and could) be deleted by both Nintendo and the creator. There was no meeting where a group sat down and decided how that configuration of lines should mean "throw the thing".

Perhaps the originator of the indicator put a note in the level description as to what it meant; over time though that indicator has become widespread. A player would play a level, determine the meaning of an indicator, and then use it in their own creations. This interaction is also intensified by the presence and popularity to Kaizo and Troll design, which explicitly foreground the persona of the designer and the player; as Wilson and Sicart's title indicates: "it's personal." If that creator's understanding of the indicator was slightly different, they might use it in a different way. As such, the meaning and usage of this language has evolved in much the same way as natural language, although its change intensified by the strong presence of the original "source language" of Mario, as well as the global "Course World", and popularity of game streaming services like Twitch.

Language and communication evolve, signs grow and change, signifiers and signifieds shift and move with use, misuse, and abuse, imitation and bad copies. The role of visual languages in enabling accessible and intuitive experiences for all users has been increasing, and it will continue to evolve with language as a whole. As visual language designers it is important to keep the intended audience firmly in mind when building an experience and a mechanism for guiding users through completing a task.

However, the intended audience might be a collection of skilled experts who've developed ways to communicate succinctly with each other. This language of indicators serves as a fun reminder couched in a video game that intention can be communicated using limited tools to leverage the audience's knowledge and skill. It can serve as a reminder to UI and UX designers that perhaps they should look at how members of the target audience are already conveying information to each other and common knowledge within the intended users. That may give the language designer a starting point to provide an experience that is familiar, more intuitive, and more comfortable to the user while leveraging their experience and knowledge to enhance how information is communicated. This can also ease the burden on the designer, as much of the design work may already have been completed as the users allowed their communication to change, allowing more time for implementation, testing, and refinement.

Visual language in *SMM* also serves to re–emphasize the human and linguistic dimensions of digital interfaces; regardless of the bits, bytes, and bots at work, the practice of language, visual or written, digital or analog, is about connecting with other people. That is, far from the negative connotations it has "abusive" game design requires the same kind of empathy for the player as the "'user-centered' approach...[that] places considerable emphasis on engineering particular kinds of challenges for the player" [7]. The difference between abusive game design and "good" design is a difference in audience, rather than a quality inherent in the games themselves. After all, even challenging, brutal, and dishonest levels, like Trolls, can and are enjoyed widely because they satisfy the expectations and needs of their user; whether Kaizo, Troll, or Regular, a level is "bad" because it doesn't meet those expectations, different as they might be for different audiences.

References

[1] J. Blinn, Floating-point tricks, IEEE Computer Graphics and Applications 17 (1997) 80–84.

[2] P. D. Bitonto, T. Roselli, V. Rossano, E. Frezza, E. Piccinno, An educational game to learn type 1 diabetes management, in: Proceedings of the 18th International Conference on Distributed Multimedia Systems, DMS 2012, August 9-11, 2012, Eden Roc Renaissance, Miami Beach, FL, USA, Knowledge Systems Institute, 2012, pp. 139–143.

[3] E. J. Marchiori, Ángel del Blanco, J. Torrente, I. Martinez-Ortiz, B. Fernández-Manjón, A visual language for the creation of narrative educational games, Journal of Visual Languages and Computing 22 (2011) 443–452.

[4] J. Quijano, The Composition of Video Games: Narrative, Aesthetics, Rhetoric and Play, McFarland, 2019.

[5] A. R. Galloway, Protocol, Theory, Culture & Society 23 (2006) 317–320.

[6] I. Lefebvre, Creating with (un) limited possibilities: Normative interfaces and discourses in super mario maker, Loading... The Journal of the Canadian Game Studies Association 10 (2017) 196–213.

[7] J. Newman, Kaizo mario maker: rom hacking, abusive game design and nintendos super mario maker, Convergence 24 (2018) 339–356.

[8] Nintendo, Nintendo Terms of Use – Official Site, https://www.nintendo.com/terms-of-use/, 2016. Accessed: 2022-28-07.

[9] D. Wilson, M. Sicart, Now it's personal: on abusive game design, in: Proceedings of the International Academic Conference on the Future of Game Design and Technology, 2010, pp. 40–47.

[10] M. R. Johnson, Playful work and laborious play in super mario maker, Digital Culture & Society 5 (2019) 103–120.

[11] E. P. Kumpf, Visual metadiscourse: Designing the considerate text, Technical communication quarterly 9 (2000) 401–424.

[12] F. De Saussure, Course in general linguistics, Columbia University Press, 2011.

[13] Eurogamer, Miyamoto on World 1-1: How Nintendo made Mario's most iconic level, 2015. https://www.youtube.com/watch?v=zRGRJRUWafY.

[14] S. Baird, Super Mario Maker 2 Players Have Uploaded More Than 26 Million Levels, https://screenrant.com/super-mario-maker-2-upload-26-million-levels/, 2021. Accessed: 2022-28-07.

[15] FANDOM, Carl's Troll Contests | Mario Maker Trolling Wiki | Fandom, https://mario-maker-trolling.fandom.com/wiki/Carl%27s_Troll_Contests, 2020. Accessed: 2022-27-07.

[16] T. R. Green, Cognitive dimensions of notations, People and computers V (1989) 443–460.

[17] D. Moody, The "physics" of notations: Toward a scientific basis for constructing visual notations in software engineering, IEEE Transactions on Software Engineering 35 (2009) 756–779.

[18] C. Shackell, Finite Semiotics: A New Theory of Semiotics With Applications To Information Technology, Ph.D. thesis, Queensland University of Technology, 2019.

[19] S. Chang, M. Barnett, S. Levialdi, K. Marriott, J. J. Pfeiffer, S. L. Tanimoto, The future of visual languages, in: Proceedings 1999 IEEE Symposium on Visual Languages, IEEE, 1999, pp. 58–61.

[20] GrandPOOBear, RACE this arrow from MAX MAP HEIGHT [SUPER MARIO MAKER 2], https://www.youtube.com/watch?v=5wQlFpLYVsg, 2021. Accessed: 2022-03-02.

[21] FANDOM, Mechakoopa | Super Mario Maker Wiki | Fandom, https://supermariomaker2.fandom.com/wiki/Mechakoopa, 2020. Accessed:

2022-03-02.

[22] jlubkinchavez, Solved: New quizzes survey workaround and complete/incomplete, https://community.canvaslms.com/t5/New-Quizzes-Users/New-Quizzes-survey-workaround-and-complete-incomplete/td-p/194757, 2020. Accessed: 2022-03-09.