Resolution and the Stack: Mechanics to Resolve Complex Actions in Card Games Over time

William Tallarico
wct9667@rit.edu
MS GDD '26
Rochester Institute of Technology
Rochester, New York, USA

Abstract

We have created a digital, dungeon crawling card game called *Index 1391.* In this game, players must explore a dungeon and interact with its denizens via a primary point of interaction called, "the stack." We have based "the stack" on a resolution system for card and board games that can be traced back to before the first collectible card game, *Magic the Gathering* (1994) made it a common design element [MTG]. Despite its history, little is known about how this constant in both board and collectible card games gained its prominence as a background resolution system. To track the mechanic, we sampled card, collectible card, and board games over time.

From these games, we can view the trend on how it first appeared, changed over time, and how we have used it to inform the central zone of interaction in our game, *Index 1391.* The stack is at the forefront of our design with the player directly interacting with its structure during the exploration of our dungeon. This novel implementation of a traditionally background system places the stack directly in front of the player. We will ask our players, how understandable is a complex resolution mechanic when it is used in the foreground of a game? In addition, we will survey how our players feel about it being used as a primary zone of interaction? As such, we will contribute both a history of the stack as well as the results of how players feel about engaging with the stack directly.

Keywords

Collectible Card Games, Inductive Design, Thematic Analysis, Rogue-likes, game design

1 Introduction

With the publication of *Balatro* (2024), the blending of card-driven gameplay, dungeon exploration, and elements of the game *Rogue* (1985 [Rog] have presented new opportunities for video game design. Card driven dungeon exploration games such as *Balatro* (2024)[Bal] are defined as roguelike deckbuilders. Roguelikes, as stated above, are games based in some fashion on the game *Rogue* (1985) [Rog]. Each game has a somewhat randomly generated world, with high difficulty, permanent death, and unpredictability [10]. Deckbuilders are defined as games in which players start with a deck which they aim to improve iteratively through gameplay, both by adding and removing cards [2].

At its core, the novelty of card-driven gameplay (often a mixture of mathematical elements that are easily transferable to computer-mediated gameplay) is nascent. Most discussions of card-drive gameplay begin and end with games played with 52 cards or the elements of *Magic: The Gathering* (1993), a collectible card game created by Richard Garfield in the mid-90s that still dominates the market [MTG]. In *MTG* (1993), players take turns which are split into phases and use a resource called Mana to cast spells (aka play cards) from their hand.

When cards are played, the opponent has an opportunity to respond and then the player of the card has a chance to respond to the response. The list of cards being responded to is referred to as, "the stack." This liminal space, this zone of resolution in is defined as "the zone in which spells, activated abilities, and triggered abilities wait to resolve." [MTG] This space follows the computer science logic of the same name [11] in that the stack follows a logical pattern of "last in, first out" (LIFO) to resolve the sometimes complex gameplay of mtg.

In the present research, we contribute to the exploration of and potential use of "the stack" in card-driven, computer-mediated gameplay. We developed a novel use of the stack in our game *Index 1391.* The development process required us to explore how the stack has evolved over time, even within mtg. Our exploration happened organically with a careful landscape of different approaches to the stack and its potential allowing us to fully understand the concept's potential as an engine through which to organize our novel dungeon exploration game with deck building elements.

The present research first delves into the background of the stack mechanic, how tabletop mechanics can be transferred to a digital medium and how the mechanic was adapted for *Index 1391.* Then it describes the methods used to evaluate its understandability, with an explanation of the instrument used and analysis approach. Then it provides the analysis derived from the data, the discussion of the results, and areas for future development.

2 *Index 1391*

Index 1391 is a 2D rogue-like deckbuilder, where the player uses cards to defeat enemies, with the end goal to defeat the "VIRUS" boss at the end of the game. The player starts with a limited deck, gaining new cards as they explore and win battles. The exploration element comes from the game's "network", a procedurally generated set of rooms that the player must explore to gain new rewards and seek out the boss.

The core gameplay of *Index 1391* is centered within a battle with an enemy. To defeat an enemy, the player must reduce their opponents health to zero. This is done through cards, although cards don't directly interact with the enemy. When a card is played, its effect(s) are added to the stack. This mimics the LIFO data structure, where recently played effects resolve first.

For example, playing a card which deals 1 damage to the enemy, would add a "protocol" to the stack that does such an action when it resolves. At the end of the turn, the stack resolves from top to bottom, and deal the 1 damage to the enemy. If a block-1 exists above the damage-1, it would block that damage, whereas if it was below the damage, it would not block the damage-1. Generally, the enemy goes first, adding protocols, then the player plays cards, followed by the stack resolving. This repeats until there is a winner of the battle.

Protocols that exist within the stack can be modified. Protocol effects that don't interact with the enemy or player modify protocols that exist in the stack in some fashion. They can either move protocols around in the stack, which will change the resolution order, or they can change aspects of the Protocols, modifying them in various ways.

The stack has limits that are intended to mimic the constraints of the memory structure from programming. The stack has a maximum capacity of protocols it can hold. Protocols themselves each have a variable size, which depends on the nature of the action that created it. Cards each have a cost, which directly relates to the size of a created protocol. For example, a card that costs 3 would create a size 3 protocol. When the stack exceeds its maximum capacity, it "overflows." Upon overflow, protocols within the stack are pushed out from the bottom to make room. To make the pressure of capacity greater, the protocols on the stack persist between turns, which means each played card has a chance to have its effect happen multiple times across rounds of game-play.

3 Background

In our design, we used an inductive evaluation of how different card games, deck building games, and other kinds of games used the stack or similar mechanics. In order to understand our exploration, we move from the use of inductive reasoning in design to that of the history of the games we used as our landscape of potential. We conclude by discussing how and why we used the exploration in our game, *Index 1391*.

3.1 On Inductive Reasoning in Design

To properly inform design of the stack, we had to understand how the stack came to be, and what games over time expanded upon the mechanic. This research intends to catalog usage of the stack from its earliest implementation within games such as *MTG* (1993) [MTG], and tracking versions of the mechanic. Such a exploration will assist in the development of *Index 1391*. Games are constantly releasing, meaning this will be in no way a comprehensive analysis of the entire history of the stack, but instead a focused look at its development over time. We recognize that since games are constantly releasing, this will not include every game that contains the affordances presented within the stack or similar mechanics [23].

While databases of games exist, none do so that quantify the stack mechanic specifics we are looking for. An "exhaustive representation is simply not possible" [12]. The stack has been background mechanic, and even within *MTG* (1993) [MTG] it is rarely referred to on cards or by designers due to its understandability [20]. Action resolution structures within games often are background systems that are implied, with rules for resolution built into individual cards. This is the reason why many game rule systems specify that a card ruling precedes a game rule. In *MTG* (1993) [MTG], rule 101.1 states that "whenever a card's text directly contradicts these rules, the card takes precedence" [18]. The background nature of the stack creates some difficulty within finding similar affordances to those that exist within *Index 1391*. As a result of this difficulty, tracking the stack and similar mechanics historically is the approach of this research.

The history of the stack within games is vast, and a large frequency of games exist which use some form of the complex resolution structure. Rather than search existing databases for reference to the stack, a more organic approach will be used, where games are selected on personal knowledge and supported through purposeful sampling principles [19]. Purposeful sampling itself represent a technique used within qualitative research that is not random, each sample is intentionally selected [19]. A majority of this work is built upon our knowledge within games and gaming to create 2 Ludographies (1 Digital and 1 Analog) with the focus of our research.

The collection of games used within this research is not comprehensive or exhaustive. It uses the landscape of existing designs that use the stack [3]. A landscape refers to a method of sampling which aims to produce a comprehensive yet focused group of samples, basing new additions off of inclusion or exclusion criteria, often using novelty [3]. Such sampling has been done to collect games based on mechanics before [12, 21].

The games themselves will be provided with an abbreviation that represents the games name, for instance *Magic: The Gathering* will be [MTG]. The landscape[3] of games has conditions to allow inclusion or exclusion from the landscape. The base approach is the novelty of the entry. The considered games for the landscape are derived from tabletop games and their systems. This does not exude games that have had a digital game created with the same sets of rules.

3.2 The Stack as a Programming Data Structure

The stack is a last-in-first-out data structure within programming, historically referred to as a *pushdown store* [11]. It was first referenced by Alan Turing in 1937 [24]. Within the stack, newer items are placed above older ones [14]. The stack has basic operations that are consistent across their implementations. Each stack has the operations *push*, *top*, *pop*, *isEmpty* [11, 14].

Within computers, the stack is used more than just the a data structure. It is also used for memory management within computer architecture using the aforementioned data structure [6]. Programs use the stack to allocate some of the memory used by programs. Stack based allocation is fast and allows for easy memory management [6]. The stack has a limited size which limits how much it can be used; when too much of the stack is used a stack overflow can occur, which often crashes the application [6].

The stack as a programming structure has influenced many game affordances and influenced the mechanical design of *Index 1391* to a significant degree. Within *Index 1391* the stack overflows when its capacity is exceeded, much like a memory stack. The manner from which our stack resolves effects is top to bottom, which mimics the data structure's implementation.

3.3 Collectible Card Games, Deck-building games, and their compatibility with Video Games

3.3.1 CCGs and the Stabilization of the stack. *Magic the Gathering* (1993) [MTG] popularized the stack with its release in the mid-90s. It was developed to assist with game actions that broke the standard [16] turn based continuity pattern [22]. The stack within *MTG* (1993) [MTG] main's function is to "keep track of abilities and triggers within the game so that they can be executed in a predictable order" [16]. The stack within *MTG* (1993) [MTG] has changed in form and ruling since it was released in 1994. Magic started with a complex resolution structure called the batch. Generally, the batch functions in a stack-like manner. Cards resolve in a LIFO manner.

The batch had extra complexity, mainly due to an "interrupt stack" [13] which had faster effects called interrupts that would create separate stacks that could not be interacted with by non-interrupts [9]. [1] Additional timing rules were also enforced, which added differing rules depending on the type of effect. This is personified in the damage prevention step, which assigned damage after the full batch [2] resolved, opening a window for another batch to respond to the incoming damage. The result of this structure was a convoluted resolution sequence, from which multiple batches could be created and resolving due to one set of card plays.

Beginning with the 6th [3] edition of *MTG* (1993) [MTG], the resolution structure began to stabilize. Rather than use a batching system, spells and abilities were added to the stack, a LIFO structure that resolved when both players passed priority [5]. Interrupts, which acted as a "faster than fast" resolved like normal instant cards and did not generate a new stack [17]. Finally, additional steps such as damage and its prevention step were made obsolete, and damage applied immediately [5]. The result of these rule changes were a much simpler system, one which acted much like the programming stack. There was one stack, rather than multiple batches. This stabilization continue through the 8th edition, where minor improvements were made in terms of what gameplay steps were added to the stack [4]. Since then, the stack has remained stable, and relatively unchanged in implementation. Since *MTG* (1993) [MTG] started in 1993, the stack resolution structure took a decade to solidify.

Other games within this time period had similar affordances to the *Magic* [MTG] stack. Before *magic* was released in 1993, the game *Robo Rally* (1994) [RR] made by Richard Garfield included stack-like resolution even though it published later than *MTG* (1993) [MTG]. In *Robo Rally* (1994) [RR], multiple players would give instructions to robots in a turn order which would resolve in a last-in-first-out manner.

After the release of *MTG*(1993) [MTG], many additional collectible card games released with similar resolution structures to the stack. The games *On The Edge* (1994) [OTE] and *7th Sea: No Quarter* (1999) [7SNQ] had cards that could react to other actions in the game, performing their effects before the original effect resolved. *On The Edge* (1994) [OTE] also had instant-like cards called *whammy* cards, which had similar resolution structure to the magic stack. *Yu-Gi-Oh* (2002) [YGO] had what was called a *chain*, structurally similar to mtg, the chain cannot be interacted with during its resolution, yet allows for cards to have different play speeds.

Netrunner (1996) [Net] and its later version *Android:Netrunner* (2012) [ANet] had a priority system for *paid* abilities within the game, where players can respond to another with as many actions as they wish until they choose not to. *Pokemon* (1996) [Pok], which came out in 1996, did not use any sort of stack resolution structure, there are triggered effects but they cannot be responded to. The result of this led to *Pokemon* (1996) [Pok] having less interaction on other player's turns, which lends itself to simpler yet less interactive game-play.

3.3.2 Deck Building Games. Most deckbuilding games, which use some form of deck iteration and expansion [2], have limited formal structures like the stack, yet often have effects that can change timing, such as scrap triggers in *Star Realms* [SR]. One of the most prominent early deckbuilding games is *Dominion* (2008) [Dom]. *Dominion* (2008) [Dom] is a deckbuilding game designed by Donald X. Vaccarino.

To win, players must accumulate the most victory points before the game ends. Each turn, players use cards to gain money, take actions, and buy new cards from a shared supply. All the money, victory, and actions the player collects and use come from their deck. The game itself ends when card piles in the supply are bought. In relation to the stack itself, dominion has reactions, which can respond to specific game actions. Cards like *Moat* allow the player to reveal the card in response to an attack, neutralizing the effect of the attack. *Dominion* (2008) [Dom] has additional relevance due to the deckbuilding structure of *Index 1391*.

The *Lord of the Rings: The Card Game* (2011) [LOTR] is a living card game. Living card games operate much like trading card games, with evolving and continuous expansions, the main difference is model of expansion. LCG's operate with a "what you see is what you get" approach meaning each player knows exactly what cards they get from each product [8]. This differs from the random distribution models of other card games. In the *Lord of the Rings: The Card Game* (2011) [LOTR], a pile of cards is used to denote a deck of encounters, from which waves of enemies attack the players. The player cooperatively fight this challenge deck and must complete quests to win while under duress. The game itself does have stack-like triggers, where *responses* can respond to specified actions, actions can be used at almost any time, and *event* cards are instantaneous [7]. These represent potential event chaining during specific windows of gameplay.

3.3.3 Other kinds of Games. Other games use stacks of cards to represent game states. The game *Escape the Dark Castle* (2017) [ETDC] uses a stack of rooms to build the dungeon. This mimics the a LIFO structure as each room has one or more effects. This stack of rooms pops each round, with the players resolving that room card and performing its actions. Furthermore, during each room, players each roll dice to beat encounters or attack [ETDC]. Players can order each other however they wish, allowing for manipulation

[1] Old *MTG* (1993) [MTG] rulebooks can be found here: https://hudecekpetr.cz/old-magic-rules
[2] A useful diagram of the batch system can be found here: https://premodernmagic.com/blog/re-introducing-the-batch/
[3] 6th Edition rules changes can be found here: https://mtg.fandom.com/wiki/Sixth_Edition/Rules_changes

of a “stack” of attacks, where each player would perform their attacks in a chosen order for an optimal outcome.

The Grizzled (2015) [TG], a cooperative card game, uses a similar system to the *Lord of the Rings: Living Card Game* (2011) [LOTR] and *Escape the Dark Dungeon* (2017) [ETDC], it uses a deck of hazards that the players must complete to win. If a certain number of the same hazard are drawn, the game is lost, yet certain effects can cancel or respond to hazards when they have been drawn.

3.3.4 Application to digital. Card game mechanics tend to be easily transferrable to a digital medium. According to Altice [1] card games are broken down into 5 key elements, planar, uniformity, ordinal, spatial and textural. Planar refers to the 2 planes that determine the card, a front and a back. Uniformity means that cards "share a similar profile" [1]. Ordinal is derived from planar and uniform, which allow cards to be grouped into sets and ordered, commonly seen within playing cards [1]. Spatial is a cards ability to occupy space in the world, and textural, the physical manipulation of cards by players [1].

Textural is the only element which struggles within a digital space, and developers often attempt to mimic physical elements within card games to mimic the lack of physical texture. The game *Balatro* (2024) [Bal] uses lots of animations to generate texture though a screen. Physical card games have lots of "manual processes" that are physically limited, these processes are exactly what computers excel at [1]. *Magic: The Gathering* [MTG] itself has computation complexity that rivals programming languages, with the structure that "mimic writing, compiling, and executing a computer program" [15]. Its mechanics were easily transferrable to digital, which resulted in multiple digital games such as *Magic: The Gathering Arena* (2019) [MTGA] and *Magic: The Gathering Online* (2002) [MTGO].

A computer also allows for greater complexity within a game’s mechanics. A card game physically is constrained by the player’s ability to perform calculations and manipulate cards. Within a computer, this is all automated, speeding up gameplay. Numbers are often an indicator for this, the game *Slay The Spire* (2017) [Slay] uses larger numbers in comparison to standard 52 deck of playing cards. *Card Wars* (2014) [CW] has "wildly inflated" statistics within its digital version compared to its analog [1]. *Hearthstone* (2014)[Hea] leverages randomness that would otherwise be tedious to calculate physically. Even with a direct adaptation to digital, a digital medium can reduce the load on a player and increase the understandability of a game. *Magic: The Gathering Arena* (2019) [MTGA] limits incorrect plays which "makes the complex play of *MTG* (1993) [MTG] more approachable" yet fundamentally changes the gameplay experience from that of an analog space [15].

In essence, card games and their mechanics are easily transferrable to a digital medium, with potential advantages due to the power of computers and the ability to offload computation from the player. There exists player experience differences between analog and digital, with digital gameplay changing the player’s interaction with the gameworld to that of a screen. The digital mediums ability to reduce computational load on the player also fundamentally changes interaction with a mechanic. The differences should be acknowledged as they could both improve and damage a player’s experience within a game depending on the contexts present.

3.4 Applying this background to our game

The exploration of games that used similar affordances to those in *Index 1391* inspired many aspects of its mechanical design. The stack, being the main interaction point within the game, is derived from a combination of multiple games implementations in the exploration. *Magic: The Gathering* (1993) [MTG] uses the stack in a similar manner to *Index 1391* as effects are added to such a stack, that resolves in a last-in-first-out manner. Also like in *MTG* (1993) [MTG], items in the stack can be changed.

The ability to move items that exist within a stack is a derivation of *Magic’s* manipulation of the stack, and operates much more like a deck of cards with the operations that can be done. The differences happen from the timing, as not every item in the game is based around stack resolution. Cards add protocols to the stack, but unlike *MTG* (1993) [MTG], the cards themselves are not resolved within a stack, and their effects happen linearly when played like other deckbuilder games such as *Dominion* (2008) [Dom]. Furthermore, triggers, a feature of many of the games are implemented outside of the normal stack mechanic. Certain protocols on the stack could have an effect when something triggers it, such as a card play.

In terms of the cards themselves, there exist 2 types within *Index 1391*, those which have a delayed effect that add a protocol to the stack, and those that instantly have that effect. Many of the games used cards that had an effect that happened when the card was played. This mechanical difference was present in *Dominion: Seaside* (2009) [DomS], where certain cards would have an effect on the start of a player’s next turn.

The stack as a data structure also heavily inspired the stack mechanic as well as many of the discussed games. The limit on the size of the stack is a feature of the memory stack in programming, this hard limit on size also exists within *Robo Rally* (1994)[RR] yet there is no overflow mechanic. The overflow mechanic was derived directly from the stack overflow error within programming when the stack’s maximum size is exceeded.

4 Methods

4.1 Prototype

To test our game, a paper prototype was created from our exploration. This prototype assisted development for our first development milestone, and playtesting soon took advantage of a digital version. The paper prototype itself allowed for rapid testing of various mechanical implementations of the stack, cards, enemies, and even deckbuilding progression. Physically, the paper prototype was made up of printed cards, 3d-printed pieces, and dice. Each of these components was used to represent the stack, elements within the stack, and the cards the player played with.

For each playtest, the player would engage in one or more battles with a deck, adding new cards after each battle to modify gameplay. For the paper prototype specifically, a moderator was present to provide and explain the rules of the game. This explanation process varied depending on the moderator, which sometimes led to inconsistent rules explanations. In addition, the moderator acted as the computation behind the game, and would manipulate game pieces as a computer would in reaction to player inputs. The moderator also acted as the primary enemy in the game, acting under strict rules for the actions they could take, and processes they had to follow.

Early in development, the prototype became tedious to maintain as mechanics were refined. While the prototype was in use, the game was built using Unity game engine, and playtesting soon transitioned to the digital medium. When moving to the digital prototype, players were provided minimal context when playing the game. The rules and objective were not explained, and present moderators were instructed to only take notes and observe gameplay.

4.2 Instrument

For the purposes of this study, we asked playtesters to fill out a questionnaire regarding their play experience after playing a playthrough of the game, either through digital or analog (subsection 8.1). The first section of the instrument addressed the respondents experience within programming and card games. It did so through asking about number of years the respondent had programmed, how experienced they were regarding the stack data structure.

Then, the respondents experience with deckbuilding and collectible card games was measured, as well as what recent deckbuilding an ccg games the respondent had played if any. After the section regarding experience, the respondent had to fill out Likert questions regarding the understandability of the stack and related mechanics, as well as the strategy and experience involved. The final section of the survey had multiple free response questions, gathering various feedback from the respondents relating to the game design and items the respondents wished to change.

Respondents were recruited from playtesters of the game, which gathered at official Capstone playtests, as well as team hosted playtests. They would be asked after playing if they wished to take a survey regarding their experience. They could opt out and choose not to complete the survey at any time.

4.3 Method

The survey had a total of 18 respondents across 4 playtests. As shown in Figure 1, the respondents were distributed fairly evenly, aside from the final playtest which only had 1 respondent. Each playtest had a different prototype of the game, with the first being a paper prototype and the rest being digital variants.

To evaluate if experience within programming and games had an impact on understandability and strategy, we mapped the levels of experience with each question, averaging the respondents values per level. We also took the overall averages to track base understandability across the questions. Finally, we also tracked the version of the prototype used, to explore if that impacted respondent perceptions of understandability or gameplay.

5 Analysis

5.1 Understandability

The average understandability was above average across all responses. The average understandability of the stack (SSENCE) was 3.83 out of 5. Stack resolution (SRESSENCE) had an average of 4.28, and protocols (PROTSENCE) had an average of 4.06. All of these averages are above neutral (3). This trend stayed relatively stable

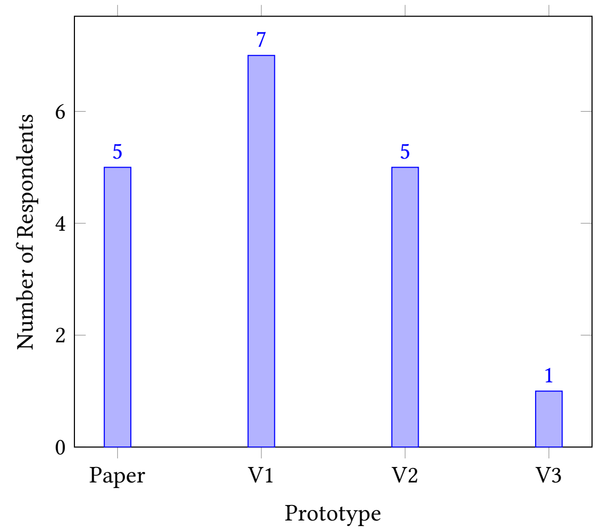

Figure 1: Respondents for each Prototype version

across playtests, with values remaining above average for each prototype with the exception of V3, which had only one respondent (Figure 3). Stack understandability (SSENCE) and stack resolution understandability (SRESSENCE) both increased in understandability moving from paper to digital, while protocol understandability (PROTSENCE) dipped.

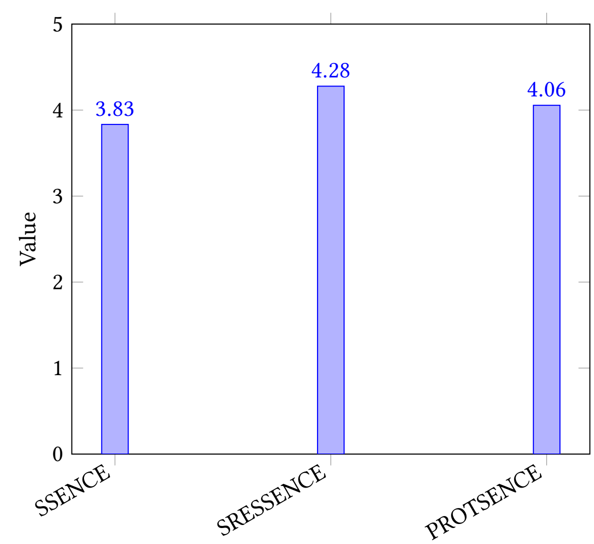

Figure 2: Average Understandability

5.2 Experience Levels

There were no significant trends across levels of experience within deckbuilding and collectible card games. Greater experience in either department did not have any significant connection to greater understandability of the game, or differences in perceptions of

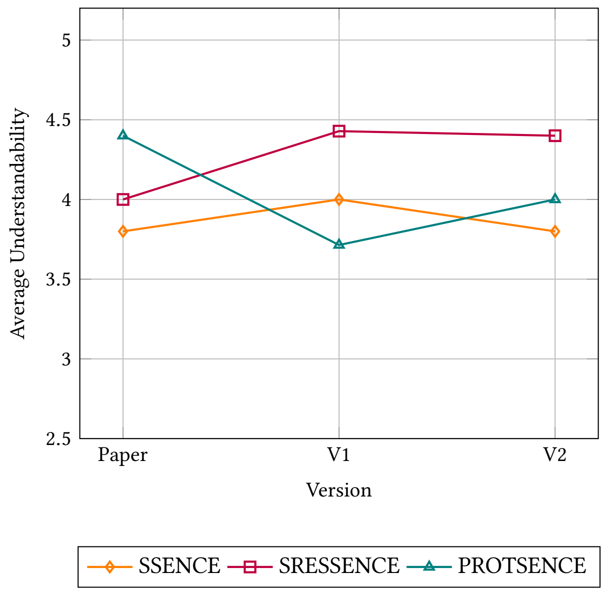

Figure 3: Understandability trends across versions showing changes from Paper prototype to V2.

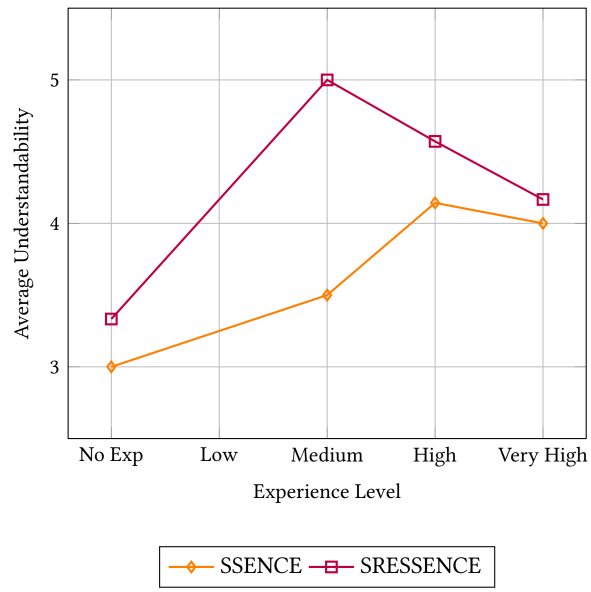

Figure 4: Effect of experience level on SSENCE and SRESSENCE answers. No respondents answered with Low Experience

gameplay. For programming experience, only 3 respondents had less than 2 years of programming experience with 2 of those having none. Despite trends existing that show the game could be more understandable for greater programming experience, not enough data is present to draw any significant conclusions.

Experience with the stack data structure within programming did see changes to understandability. As seen in Figure 4, the greater the experience with the stack data structure, the greater the understanding of the stack, with the exception of very high experience. For stack resolution, the understanding peaked at medium, yet dropped the more experience a player had with the data structure.

5.3 Stack Overflow Over Time

Perceptions of the stack overflow strategy changed across prototype versions. As seen in Figure 5, respondents perceived that stack overflow added meaningful strategy less for V1 of the prototype in relation to the other versions of the prototype. This stands out in contrast to the other gameplay questions, which all held fairly consistent averages across versions.

6 Discussion

6.1 Understandability

The game was fairly understandable to players of various deckbuilding and collectible card game experience. No clear trends emerged from levels of deckbuilding and ccg experience in relation to understandability. The average understandability was 3.83, 4.28 and 4.06 for the stack, stack resolution, and protocols respectfully. These are all above average, with the stacks' immediate understandability being slightly lower than the other two. A vast majority of respondents had over 2 years of programming experience. Given the small sample size of respondents with less programming experience, no concrete conclusion can be drawn from this, yet generally the game appears to be understandable to programmers if not generally understandable. In addition, the above average understandability is significant as the prototypes tested had no dedicated tutorial sequence, so the high understandability came from gameplay affordances.

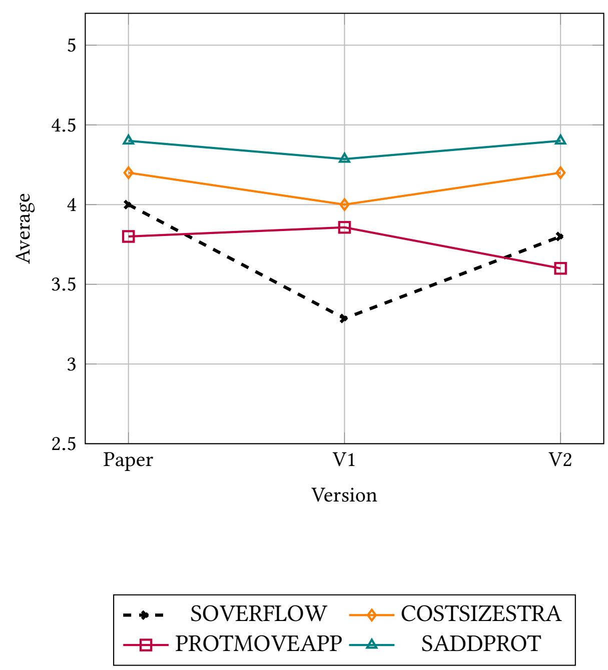

Figure 5: Gameplay metrics across versions (Paper to V2).

Furthermore, the digital prototype generally increased understandability of the game as shown in Figure 2. For stack and stack resolution understandability (SSENCE, SRESSENCE), the average understandability increased moving from paper to digital. The opposite happened for protocols, which decreased in V1 and went back up for V2. This could potentially be attributed to the paper prototype affording clearer explanation for how protocols operate with a moderator present, whereas within the prototypes user interface was in various stages of development, and could have hard a harder job conveying information.

The analysis and discussed reveal need for future work expanding the population tested. Within capstone playtests, especially at a technical institute, there exists a high population of students that program. Expanding population would allow for greater representation of playtesters outside of the realm of programming, which could test any conclusions regarding the game's understandability.

6.2 Stack Experience

When it comes to experience with the stack data structure, interesting trends emerge. For stack resolution (SRESSENCE), as experience level goes up from medium, understandability goes down. This trend is also present for the normal stack understandability (SSENCE), yet only for the jump from high to very high experience. For both, understandability was lowest with no stack experience, so a greater experience with the stack data structure does correlate to greater understanding of the game, yet only to a certain point. The stack within *Index 1391* operates much unlike a stack in programming in many aspects, the stack does not "pop" or remove elements from the top as it resolves, and effects can move protocols around in the stack, even when it is actively resolving. This breaks the norms associated with the data structure within programming. The stack can also overflow, and doing so will push elements out of the bottom of the stack, which makes it act just as a queue data structure, which could lead to further dissonance. The various disconnects between the game mechanic and how the stack operates within programming could contribute to less understandability of the stack the greater the experience with the stack data structure.

6.3 Stack Overflow

Stack overflow had a substantial dip in meaningful strategy between the paper and first digital (V1) prototypes. Mechanically in both cases, the stack overflow mechanic was identical. All protocols on the stack were removed when the stack's max capacity was exceeded. This presents the main difference between each to be the medium of gameplay. With a moderator present, the stack overflow process could be explained, yet with no moderator, the stack would process overflow and remove all protocols without explanation or clear feedback. This likely caused players to not understand how to take advantage of the mechanic, as they could not properly understand it once initially brought to a digital medium.

Stack overflow rebounded in mechanical strategy within V2. Between V1 and V2, the mechanic was changed significantly. Rather than clear the entire stack, when the stack overflowed, it changed to a queue, with older elements getting pushed from the bottom of the stack to make room for the incoming protocols. This change, along with additional animation, user interface, and sound feedback, all contributed to the new stack overflow. Mechanically, the change could've felt more intuitive to players, aesthetically, the greater feedback provided to the player could have helped in understanding their gameplay actions.

7 Conclusion

In developing the stack mechanic, we generated a history of the mechanic through tabletop history. We established that it could be brought to a digital medium, and using the history we derived our own mechanics, and in testing them within both paper and digital prototypes found that the mechanic was understandable, especially to programmers, when brought to the forefront of a game's design. Furthermore, we discovered that a greater experience with the stack data structure was linked to a lower understanding of our game play mechanics, potentially due to the mechanic altering how the data structure operates. We also discovered that the transfer from paper to digital afforded similar understanding for some aspects of the game, yet certain areas, such as stack overflow, suffered initially from the digital prototype transfer.

There exists room to expand this work further. The population surveyed was mainly programmers, which meant that conclusions derived from levels of programming were not supported well. To validate existing trends and see what effect programming levels has on understanding, testing of the game should expand to include a variety of programming and stack experiences across more context than RIT's campus.

Acknowledgments

I would like to express sincere gratitude to all those who supported my work. Special thanks to Nicolas Lalone for lots of patience, great guidance, and amazing feedback. I wouldn't have been able to do this without him.

Thanks as well to Erika Mesh, who made me explain the research so many times I ended up understanding what I was doing.

I also also appreciate the contributions of my teammates on *Index 1391* who played lots of games with me for this journey. Finally, thanks to my family, my parents, grandparents, brother and Amanda. You all listened patiently as I explained what I was doing, and provided motivation and encouragement at every step.

References

[1] Nathan Altice. 2014. The playing card platform. *Analog game studies* 1, 4 (2014).

[2] BoardGameGeek. 2001. *Deck, Bag and Pool Building*. Retrieved December 10, 2025 from https://boardgamegeek.com/boardgamemechanic/2664/deck-bag-and-pool-building

[3] Nathan Bos, Ann Zimmerman, Judith Olson, Jude Yew, Jason Yerkie, Erik Dahl, and Gary Olson. 2007. From shared databases to communities of practice: A taxonomy of collaboratories. *Journal of Computer-Mediated Communication* 12, 2 (2007), 652–672.

[4] Community Contributors. 2019. *Eighth Edition/Changes*. Retrieved December 11, 2025 from https://mtg.fandom.com/wiki/Eighth_Edition/Changes

[5] Community Contributors. 2020. *Sixth Edition/Rules changes*. Retrieved December 11, 2025 from https://mtg.fandom.com/wiki/Sixth_Edition/Rules_changes

[6] Leo Ferres. 2010. Memory management in C: The heap and the stack. *Universidad de Concepcion* (2010).

[7] Fantasy Flight Games. 2010. *The Lord of the Rings: The Card Game – Official Rules*. Retrieved December 12, 2025 from https://images-cdn.fantasyflightgames.com/ffg_content/lotr-lcg/LOTR%20Rules.pdf

[8] Fantasy Flight Games. 2025. *Living Card Games*. Retrieved December 12, 2025 from https://www.fantasyflightgames.com/en/more/living-card-games/

[9] Richard Garfield and Wizards of the Coast. 1993. *Magic: The Gathering Alpha Rulebook*. Retrieved December 11, 2025 from https://hudecekpetr.cz/other/

rulebooks/rulebook-alpha-1993/index.html Original Alpha edition rulebook hosted online.
[10] John Harris. 2020. *Exploring roguelike games.* CRC Press.
[11] Sten Henriksson. 2011. A brief history of the stack. *HAPOC11 History and Philosophy of Computing, Ghent, Belgium* (2011).
[12] Nicolas LaLone, Phoebe O Toups Dugas, and Michelle V Cormier. 2023. A Quest?!: The Secret Life of Gameworld Punctuation. *Proceedings of the ACM on Human-Computer Interaction* 7, CHI PLAY (2023), 526–557.
[13] MicroProse. 1997. *MicroProse Magic: The Gathering User Manual.* Retrieved December 11, 2025 from https://archive.org/details/mtg_manual Scanned user manual hosted on the Internet Archive.
[14] Brad Miller and David Ranum. 2013. *Problem Solving with Algorithms and Data Structures.* The University of Auckland, School of Computer Science, Auckland.
[15] Jack Murray. 2020. More Than Just the Table: Analog Games as Computational Platforms. In *Proceedings of the 15th International Conference on the Foundations of Digital Games.* 1–4.
[16] Jack Murray. 2023. Planeswalking: Magic: The Gathering Across Analog and Digital Platforms. (2023).
[17] Wizards of the Coast. 1999. *Magic: The Gathering Comprehensive Rules (April 23, 1999).* Retrieved December 11, 2025 from https://hudecekpetr.cz/other/rulebooks/comprehensive-1999-04-23.txt 1999 rulebook hosted online.
[18] Wizards of the Coast. 2025. *Magic: The Gathering Comprehensive Rules.* Retrieved December 10, 2025 from https://magic.wizards.com/en/rules Comprehensive Rules update: 2025-11-14.
[19] Lawrence A Palinkas, Sarah M Horwitz, Carla A Green, Jennifer P Wisdom, Naihua Duan, and Kimberly Hoagwood. 2015. Purposeful sampling for qualitative data collection and analysis in mixed method implementation research. *Administration and policy in mental health and mental health services research* 42, 5 (2015), 533–544.
[20] Mark Rosewater. 2006. *On The Stack.* Retrieved December 10, 2025 from https://magic.wizards.com/en/news/making-magic/stack-2006-11-20
[21] Matthew Rueben, Matthew R Horrocks, Jennifer Eleanor Martinez, Michelle V Cormier, Nicolas LaLone, Marlena Fraune, and Phoebe O Toups Dugas. 2022. "I See You!": A Design Framework for Interface Cues about Agent Visual Perception from a Thematic Analysis of Videogames. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems.* 1–22.
[22] Matthias Steinböck. [n. d.]. Game Design Patterns in Digital Card Games. ([n. d.]).
[23] Phoebe O Toups Dugas, Nicolas LaLone, Katta Spiel, and Bill Hamilton. 2020. Paper to pixels: a chronicle of map interfaces in games. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference.* 1433–1451.
[24] Alan Mathison Turing et al. 1936. On computable numbers, with an application to the Entscheidungsproblem. *J. of Math* 58, 345-363 (1936), 5.

Digital Ludography

[Bal] LocalThunk. *Balatro.* Game [Windows], 2024.
[CW] Cartoon Network. *Card Wars.* Game [Digital Card Game], 2014.
[Hea] Blizzard Entertainment. *Hearthstone.* Game [Digital Card Game], 2014.
[MTGA] Wizards of the Coast. *Magic: The Gathering Arena.* Game [Digital Card Game], 2019.
[MTGO] Wizards of the Coast. *Magic: The Gathering Online.* Game [Digital Card Game], 2002.
[Pok] Media Factory. *Pokémon Trading Card Game.* Game [Trading Card Game], 1996.
[Rog] Ken Arnold Michael Toy, Glenn Wichman. *Rogue: Exploring the Dungeons of Doom.* Game [Unix], 1980.
[Slay] MegaCrit. *Slay the Spire.* Game [Video Game], 2019.

Analog Ludography

[7SNQ] Alderac Entertainment Group. *7th Sea: No Quarter.* Game [Collectible Card Game], 1999.
[ANet] Fantasy Flight Games. *Android: Netrunner.* Game [Living Card Game], 2012.
[Dom] Donald X. Vaccarino. *Dominion.* Game [Deckbuilding Game], 2008.
[DomS] Rio Grande Games. *Dominion: Seaside.* Game [Board Game Expansion], 2009.
[ETDC] Themeborne Ltd. *Escape the Dark Castle.* Game [Board Game], 2017.
[Epi] Wise Wizard Games. *Epic Card Game.* Game [Card Game], 2015.
[FAB] Legend Story Studios. *Flesh and Blood.* Game [Trading Card Game], 2019.
[FOW] Force of Will Co., Ltd. *Force of Will.* Game [Trading Card Game], 2012.
[LOTR] Fantasy Flight Games. *The Lord of the Rings: The Card Game.* Game [Living Card Game], 2010.
[ML] Upper Deck Entertainment. *Marvel Legendary.* Game [Deckbuilding Game], 2012.
[MTG] Wizards of the Coast. *Magic: The Gathering.* Game [Trading Card Game], 1993.
[Net] Richard Garfield. *Netrunner.* Game [Collectible Card Game], 1996.
[ODD] Asmadi Games. *One Deck Dungeon.* Game [Card and Dice Game], 2016.
[OTE] Atlas Games. *On the Edge.* Game [Collectible Card Game], 1995.
[RR] Richard Garfield. *Robo Rally.* Game [Board Game], 1994.
[SR] Wise Wizard Games. *Star Realms.* Game [Deckbuilding Game], 2014.
[TG] Sweet Games / CMON. *The Grizzled.* Game [Card Game], 2015.
[YGO] Konami. *Yu-Gi-Oh! Official Card Game.* Game [Trading Card Game], 1999.

8 Appendices

8.1 Appendix A: The Instrument

About how long have you been programming?

- I do not know how to code.
- 0-2 Months
- 3-6 Months
- 6-12 Months
- 1-2 Years
- Over 2 years

How much experience do you have with the stack data structure in programming. *(1-5 with 1 being None and 5 being Lots of Experience)*

How much experience do you have with deckbuilding games (e.g. Dominion, Dragonfire, Sentinels of the Multiverse) *(1-5 with 1 being None and 5 being Lots of Experience)*

How much experience do you have with collectible card games, trading card games, or living card games (e.g.Magic the Gathering, Lorcana, Star Wars: Unlimited, Lord of the Rings the Living Card Game) *(1-5 with 1 being None and 5 being Lots of Experience)*

What was the most recent deckbuilding game you played? If you haven't played one, please write, "n/a".

What was the most recent trading or collectible card game you played? If you haven't bought one, please write,"n/a"

As a reminder: The stack is a gameplay zone in Sanctum Protocol where player and enemy actions (protocols) are resolved. It resolves at the end of your turn from top to bottom, protocols within the stack stay in the stack for future turns.

Understanding and Gameplay Questions *(1-5, with 1 being Strongly Disagree and 5 being Strongly Agree)*

- The stack in Sanctum protocol immediately made sense.
- At the end of your turn, the stack resolved, performing the effects of the protocols in it. The resolution of the stack made sense?
- The protocols (actions added to the stack by cards and the enemy) made sense.
- Did the stack Overflow mechanic add meaningful strategy to your experience
- (This is when the maximum size of the stack was exceeded forcing it to be cleared, doing damage to the entity (player or AI) that overflowed the stack).
- Did making card costs equivalent to the size of protocols in the stack make the game feel more strategic
- How the protocols moved or were modified within the stack feels appropriate.
- Adding protocols to the stack added to the experience of the game.

Open Ended Questions

- What was your strategy to win the game?

- If you could change just one thing, what would it be?
- Is there anything you feel was missing from the game?
- Anything else you would like to comment about the game?