

Less: the Praxeology of Minimalist Game Design

Marc-Antoine Jetté-Léger

A Thesis
In the Department of
Design and Computation Arts

Presented in Partial Fulfillment of the
Requirements for the Degree of Master of Design
at Concordia University
Montreal, Quebec, Canada

November 2025

© Marc-Antoine Jetté-Léger, 2025

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis

Prepared By: Marc-Antoine Jetté-Léger
Entitled: Less: the Praxeology of Minimalist Game Design

And submitted in partial fulfillment of the requirements for the degree of the

Master of Design

Complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final Examining Committee:

_____ Chair
Jonathan Lessard

_____ Examiner
Rilla Khaled

_____ Examiner
Pippin Barr

_____ Thesis Supervisor
Jonathan Lessard

Approved by _____
Pippin Barr, Chair of Department

Annie Gérin, Dean of Faculty

Abstract

Less: the Praxeology of Minimalist Game Design

Marc-Antoine Jetté-Léger

This thesis investigates minimalism as an aesthetic and a praxeological framework for game design through a research-creation process centered on the iterative development of a minimalist single player puzzle videogame.

Through many iterations of design, testing, and reflection, features like procedural level generation, visual feedback systems, animation timing, and interface simplification were examined in relation to Upton's heuristics for playfulness (choice, variety, consequence, predictability, uncertainty and satisfaction) and a proposed set of minimalist heuristics (narrow decision spaces, procedural generation, abstraction and wholeness). Each feature acted as a case study in how design and minimalist constraints can influence the player's capacity to understand, experiment and engage with the game's system.

Points of convergence and of tension emerged, such as abstraction versus predictability or wholeness versus variety, revealing that minimalist game design is not defined by reduction alone, but by carefully calibrating of what is available for interaction, what is shown and what is left for the player to infer.

In the game design process, minimalism acts as a tool for dynamic balance between clarity and mystery, and between coherence and affordance. The results of this research demonstrate how minimalist game systems can sustain playfulness and player engagement by turning limitation itself into generative design principles.

Table of Contents

List of Figures	vi
Introduction	1
Upton's Six Heuristics for Playfulness.....	3
Choice	3
Variety	4
Consequence	5
Predictability	6
Uncertainty	7
Satisfaction.....	8
Four Heuristics for Minimalist Game Design.....	9
Narrow Decision Spaces	9
Procedural Generation	10
Abstraction	11
Wholeness	12
Synthesis.....	13
Methodology	14
Iterative Game Development	14
Process Documentation and Data Collection.....	15
Thematic Tagging and Heuristic Mapping.....	16
Synthesis and Reflection.....	17
Process, Analysis and Results.....	18
Game Concept and Core Mechanics	18
Minimal Playable System	19
Expanding the Design Space	20
Heuristics for Play and Minimalism	20
Iterative Feature Development.....	22
Clickable Squares in an Array	22
Squares Toggling Other Squares.....	24
Goal Arrangement of Squares	26
No Trivial Solutions	27
Changing the Shape of the Goal Array	28
Completing and Generating New Levels.....	29
Undo, redo and reset functionalities.....	31
Clicks Restriction.....	33
Self-Target Indicator Change	34
No Pre-Solved Levels	35
Longer arrays	36

Prediction Rectangles	37
No Useless Arrows.....	38
Difficulty Through Click Counts	39
Undo, Redo, and Reset Available After Level Completion.....	39
Multi-Solution Mode	41
Wrap-around Toggles	43
Enforcing Number of Solutions	44
Progression.....	45
Variation in the Solution Found Animation	47
Cascading Squares.....	48
Cascading Squares into the Progression.....	51
Single-Solution as the Core Game Mode.....	53
No All-Self-Targeting Levels	53
Removed the Trophy Icon.....	54
Prediction Dots.....	55
Visual Punch Animations	57
Hiding UI Elements	57
Changing the Position of the Goal Array.....	59
Crosses and Checkmarks	60
More Visual Punch Animations	61
Nudge Animations and Delays.....	62
Removal of Prediction Dots and Crosses and Checkmarks.....	64
Goal Array as Squares.....	66
Points of Convergence and Tension	68
Convergence.....	68
Tension	70
Synthesis.....	71
Future Directions.....	72
Bibliography	74

List of Figures

Figure 1, <i>Toggle</i> , commit #41ff233	19
Figure 2, <i>Toggle</i> , commit #85ce024	23
Figure 3, <i>Toggle</i> , commit #0186e50	25
Figure 4, <i>Toggle</i> , commit #38ea59d	27
Figure 5, <i>Toggle</i> , commit #a8e3268	29
Figure 6, <i>Toggle</i> , commit #9670c92	31
Figure 7, <i>Toggle</i> , commit #dadcbbb	33
Figure 8, <i>Toggle</i> , commit #041ca67	34
Figure 9, <i>Toggle</i> , commit #d90dad8	35
Figure 10, <i>Toggle</i> , commit #6a44792	36
Figure 11, <i>Toggle</i> , commit #8b65ce9	38
Figure 12, <i>Toggle</i> , commit #1337c7b	40
Figure 13, <i>Toggle</i> , commit #f1cfe9e	42
Figure 14, <i>Toggle</i> , commit #683a0b6	43
Figure 15, <i>Toggle</i> , commit #06febfd	45
Figure 16, <i>Toggle</i> , commit #f32b758	48
Figure 17, <i>Toggle</i> , commit #f32b758	48
Figure 18, <i>Toggle</i> , commit #370f9ad	49
Figure 19, <i>Toggle</i> , commit #23f72f9	55
Figure 20, <i>Toggle</i> , commit #478c8ff	56
Figure 21, <i>Toggle</i> , commit #28d3215	58
Figure 22, <i>Toggle</i> , commit #c47288a	60
Figure 23, <i>Toggle</i> , commit #1f3e075	62
Figure 24, <i>Toggle</i> , commit #41ff233	67

Introduction

As video games have become a significant cultural and artistic medium, the exploration of different design paradigms emerges as an essential area of study. Among these, minimalist game design remains underexplored.

In a market oversaturated with offers for the maximum number of features, maximum complexity and maximum play time, minimalist games offer an alternative approach for game designers seeking to create unique connections with players. Minimalist games like *Tetris* (Pajitnov, 1988), *Limbo* (Playdead, 2010), and *Superhot* (Superhot Team, 2016) have demonstrated the power of minimalism in game design. Despite their impact, existing literature on their design fails to capture the breadth of design possibilities in this approach.

Throughout my years designing experimental video games, I have consistently emphasized minimalism as both an aesthetic and interactive principle. This thesis formalizes that understanding by examining how minimalist design spaces and creative constraints shape one another throughout the design process. It explores how minimalist game design can operate as a meaningful and intentional practice within the broader field of game development.

To guide this investigation, I ask: How do minimalist design constraints shape and influence decision-making in the iterative game design process? A secondary question follows: What areas of convergence and tension emerge when these constraints overlap, and how do they affect design choices?

To address these questions, this study draws on research in game design and interaction design, as well as insights from Minimalism as an art movement. It examines how foundational design principles establish constraints in game creation, and how those constraints, in turn, open new creative possibilities within minimalist game design.

There is tension between minimalism and game design. Game design traditionally values expansion and accumulation, while minimalism values reduction and restraint. This tension between the desire to build complexity and the intent to simplify defines the space in which this research takes place.

Games are complex systems of rules that create a space for meaningful play. To understand what makes a game situation playful, Brian Upton (2015) proposes six heuristics: choice, variety, consequence, predictability, uncertainty, and satisfaction. I have chosen to use these heuristics as a lens to highlight the base principles of game design because they were developed by Upton to capture the essential aspects that make play meaningful. According to him, they do so by highlighting how players act intentionally, perceive outcomes, and experience engagement through play.

Minimalism operates on different principles, ones that sometimes align with and sometimes challenge those identified by Upton. Rather than expanding systems, minimalism reduces them to their essence, simplifying mechanics, limiting decision spaces, and stripping away representation to emphasize direct, abstract interaction. Despite the emergence of minimalist games, there is still no academic consensus or formalized framework defining what constitutes minimalist game design.

To address this gap, I propose a complementary set of heuristics: narrow decision spaces, procedural generation, abstraction, and wholeness. This proposed set of heuristics is derived from readings in game design and interactive design theory and research on Minimalism as an art movement. These principles are not intended to be interpreted as definitive rules for minimalist game design but as conceptual tools for understanding how minimalist constraints can structure engagement and play through limitation.

The decision to settle on these heuristics as an analytical lens for minimalist game design did not precede the work described in this thesis. As I explain in more detail in my methodology, they emerged after development, during a systematic rereading of my design notes, journal, and documentation. After reflecting on recurring themes across this corpus, I began to see strong connections between the practical, tacit concerns that guided my making and the concepts I had encountered in the scholarly literature. The heuristics presented here crystallized during the reflective practice that followed the creative process.

Because these heuristics frame how minimalism and playfulness are understood within this project, I introduce them here to orient the reader. At the same time, they also constitute a key outcome of my inquiry. For that reason, I will return to them later to discuss their significance

and the ways they both informed and were informed by the act of creation itself.

The following section first examines Upton's six heuristics as a general framework for playfulness. It then introduces and contextualizes the minimalist heuristics I chose for this research, extracting them both from minimalist design practices in games and to the broader Minimalist tradition in art. With these two sets of heuristics, I aim to develop a theoretical foundation for understanding how playfulness and minimalism intersect in game design.

Upton's Six Heuristics for Playfulness

Here, I present Upton's six heuristics as a framework for understanding how games generate playfulness. While Upton's model provides a strong conceptual foundation, I expand on each heuristic by placing it in dialogue with other perspectives from game studies and interaction design research. This complementary approach helps in revealing how these heuristics can operate as theoretical principles for analyzing and designing playful systems.

Choice

Upton (2015) describes games as the structured interaction between a player and a system, where choices arise through constraints that define what actions are possible. He says that the action space of the game is defined by rules, interfaces, and the structure of the game world, guiding how the player moves through a field of potential decisions. According to him, players act, the game responds, and this ongoing exchange becomes play.

Zubek (2020) also frames games as systems that structure player decisions. He describes game pieces as the "nouns" of the system and rules as the "verbs," establishing how those pieces can be manipulated. Together, they define what the player can do and, therefore, what they can choose. Mechanics arise from the combination of pieces and rules, forming the foundation for the actions and the possible choices available to the players. According to Zubek, these mechanics form the basis of gameplay, which he conceptualizes through gameplay loops: repeated cycles of player action and system response. As players engage with these loops, they begin to perceive deeper layers of strategy and structure, making their choices richer and more complex over time.

Upton adds that constraints play a central role in shaping choices. He makes the distinction between external constraints, enforced by the game's physical or logical structure, and internal constraints, which players impose on themselves based on their expectations, prior knowledge, or external experience. According to him, these constraints narrow the player's decision space and the choices available to them.

Interaction design research reinforces the relationship between constraints and user choice. Norman (2013) argues that users must understand what actions are possible in order to interact effectively with a system. He says that affordances describe the potential actions inherent in an object and that constraints, whether physical, semantic, cultural, or logical, guide those actions and prevent undesirable ones. According to Norman, affordances and constraints create opportunities for choice, even when the system's internal logic is not fully visible.

According to Upton, games become playful only when they offer players meaningful choices. Yet these choices are not merely the options visible on the surface of a game. They are the product of constraints, affordances, and a deliberately constructed decision space, emerging through interaction and repeated across the gameplay loops. Upton also emphasizes the importance of balance: too many choices become unmanageable, while too few leave the game lacking in depth. The heuristic of choice becomes a tool with which to examine this balance.

Variety

Upton (2015) argues that games remain playful by refreshing the player's experience without altering their fundamental logic. He states that the variety heuristic is the game's ability to produce different experiences from the same underlying rules. According to him, choice describes the structure of player decisions, while variety concerns how those decisions lead to meaningfully different outcomes across different contexts.

This connects to Zubek's (2020) writing on the interplay of mechanics and systems. He says that when multiple mechanics interact, they can produce emergent behaviors: outcomes that no single mechanic could generate on its own. These interactions transform a constrained ruleset into a dynamic play space where new experiences arise from the combination of simpler components, layering game loops and interconnected systems.

Zubek also describes games as stateful systems defined by transitions between states. He says that a state represents the current configuration of game pieces and their properties. According to him, the full set of possible states forms the state space, while the actions available from any given state define the action space. Upton expands on this by making a distinction between the actual action space (the horizon of action) and the perceived action space (the horizon of intent): what players believe they can do versus what the rules actually allow. According to him, this horizon of intent helps players manage complexity by focusing their attention on meaningful options.

Variety emerges when these spaces shift over time. Upton says that, as states change, so do the available actions and the horizon of intent. Zubek says that not all options have the same impact, however. He makes the difference between mechanically distinct choices, which produce unique state transitions, and illusory choices, which appear different but lead to similar outcomes. Variety depends much more on the former.

As we saw earlier, Upton says that too few choices can lead to stagnation, while too many can overwhelm the player. Within his framework, variety helps maintain a balance by refreshing the configurations and situations the players encounter, even when the underlying decision space remains narrow.

In minimalist game design, where mechanics and systems are limited, variety becomes essential for sustaining playfulness without expanding the ruleset. Using variety as a heuristic invites us to consider whether the game continues to present meaningfully varied situations in which players can make choices.

Consequence

Upton (2015) introduces the consequence heuristic to highlight the moment when player decision meets system response. It is the point at which the game acknowledges a player's input and applies change to the current state. He adds that consequence is expressed through feedback, the primary source of information through which players understand the impact of their actions.

This connects with Zubek's (2020) structural understanding of games as stateful systems. He says that each action transitions the game from one state to another, and the clarity and consistency of those transitions determine how legible the system feels to the player. When transitions are understandable and consistent, players can form mental models of how the game works, experiencing consequence through coherent interactions with the system's logic.

The relationship between user action and system reaction is also foundational in interaction design. Norman's (2013) highlights the importance of clear causality, arguing that users understand systems through mappings and feedback. According to him, mappings describe the relationship between controls and the parts of a system being controlled and that natural mappings, the ones that respect spatial correspondence, temporal contiguity, and cultural conventions, are the most effective. He also says that the role of feedback is to confirm that an action has occurred and to communicate the resulting change of state.

In games, consequence incorporates all these perspectives. Player choices matter not only because they change the state of the game, but because they alter what choices become available next. Variety expands the field of possible actions, but using the heuristic of consequence can help us examine the weight and impact of those actions.

Predictability

Upton (2015) says that predictability is essential to play because it allows players to understand how the system behaves and therefore act with intention. He says that when a game communicates its available actions clearly and responds consistently, players can project potential outcomes and engage in what he calls anticipatory play. He says that this form of play involves reasoning not only from an action to its immediate effect, but through imagined future situations within the evolving system. According to Upton, this ability to anticipate conveys a useful feeling of control that supports a playful mindset.

Koster's (2004) theory reinforces this connection between predictability and meaningful play. He argues that the pleasure of play comes from learning and that players remain engaged when they encounter challenges they can overcome by applying existing knowledge while also developing new skills. In this way, predictability supports this cycle of discovery by allowing players to form, test and confirm hypotheses about how their actions influence the system.

Norman's (2013) work in interaction design mirrors these ideas. He describes how signifiers, feedforward, mappings, and familiar structures help users build accurate expectations about how systems will behave. He says that signifiers are visual, auditory, or tactile cues that communicate possible actions, making affordances perceptible. He also mentions feedforward, which provides information about the potential outcome of an action before it is taken. According to him, systems that align with existing knowledge, recognizable patterns, or cultural conventions are easier to interpret. He says that consistency reinforces learning, whereas broken conventions or inconsistent structures undermine predictability and lead to confusion. Upton echoes this by noting that frustration arises when the player's mental model diverges from the game's actual behavior.

Zubek (2020) highlights the role of fiction in games, which shapes players' expectations by contextualizing actions and clarifying their role within the game world. Fiction provides cues that help players anticipate how the system might respond, strengthening predictability.

Taken together, these ideas show how predictability allows players to learn and model a game's internal logic. The predictability heuristic helps us evaluate how games can establish a stable enough communication framework for players to experiment playfully.

Uncertainty

As described earlier, Upton (2015) distinguishes between the actual action space and the perceived action space. According to him, the gap between them generates uncertainty, prompting players to explore, experiment, and navigate the play space. He describes uncertainty as the aspects of play that remain unknown, the outcomes that cannot be fully predicted, the interactions that must be tested, and the possibilities that the player has not yet discovered

Zubek (2020) identifies multiple sources of uncertainty, including hidden information, randomization, and the combinatorial complexity that arises when multiple systems interact. He also reflects Upton's view that these forms of uncertainty create tension and sustain engagement by ensuring that outcomes are never entirely predictable.

Svanæs' (2000) work on interactivity adds another dimension through the concept of breakdowns. According to him, breakdowns occur when the system behaves in an unexpected way, prompting the user to reflect and update their understanding. He says that, while often framed as errors in traditional interaction design, breakdowns are intentionally incorporated into games to encourage players to reinterpret how the system works.

In this sense, uncertainty is not simply a lack of knowledge. Instead, it is the balance between understanding and discovery, where the system reveals itself gradually through surprise, adjustment, and recalibration. This balance is essential to play. If the outcomes of actions are entirely predictable, the player simply executes known procedures, and if they are entirely random, decisions lack meaning.

Upton says that play exists in the space between these extremes, where enough predictability allows decisions to matter, and enough uncertainty ensures that those decisions remain meaningful. We can therefore apply the uncertainty heuristic as a lens through which we can examine this balance.

Satisfaction

Upton (2015) says that satisfaction arises in play when the game makes desirable outcomes attainable. He says that whether victory is formalized through points, progress, or simply the implicit reward of "playing well," the rules must allow players to move toward more desirable areas of the game's state space. According to him, what matters is that players can make choices and pursue a distinguishable trajectory whose value can be evaluated by the player.

Norman's (2013) work provides complementary insight by framing interaction as a goal-driven process. According to him, users form intentions, take actions, perceive feedback, and evaluate results and satisfaction occurs when users' conceptual models allow them to achieve their goals reliably. He says that these conceptual models are built through affordances, signifiers, constraints, mappings, feedback, and feedforward, all of which help users form expectations and recognize when those expectations are met.

Within games, this alignment between expectation and outcome becomes central. Satisfaction as a heuristic helps us to assess the completion of the feedback cycle and how it gives players

a sense of closure, validating their understanding of the system and reinforcing continued engagement.

Four Heuristics for Minimalist Game Design

While Upton's heuristics describe what makes games playful, I have found that minimalist game design operates under different priorities. In the following section, I propose four additional heuristics for minimalism in game design, informed by game design theory, foundational principles of interactive design and research on Minimalism as an art form. These heuristics are intended as conceptual tools for evaluating how minimalist constraints shape and sustain engagement.

Narrow Decision Spaces

Although academic work on minimalist games remains limited, Nealen et al. (2011) offer useful insight into how minimalism functions in practice. They argue that minimalist games reduce a work to its most essential elements "while maintaining rich play." A key strategy for achieving this is to limit rules and narrow the decision spaces available to the player. According to them, this deliberate restriction reduces the number of possible actions to a small, clearly defined set, ensuring that every interaction is mechanically significant and directly tied to the core loop.

Nealen et al. also describe interaction as operating across different levels: macro-mechanics (high-level, human-recognizable actions) and micro-mechanics (smaller action–reaction components). For them, minimalist games often commit to a single macro-mechanic supported by only a few micro-mechanics. They also mention discretization as a technique to narrow down decision spaces. They describe discretization as the act of dividing continuous actions into discrete, more easily interpretable states. According to them, minimalist games support play by ensuring that each available option is distinct and, instead of offering a wide field of continuous actions, emphasizes a handful of self-contained interactions.

This reductionist approach introduces a tension between minimalism and playfulness. Smaller decision spaces reduce variety but minimalist games can navigate this tension not by adding new mechanics, but by inviting players to explore the expressive potential of the core ones.

Minimalist art theory reflects a similar reduction of artworks to their essential components, rejecting complexity and sophistication as a measure of quality. Judd (1964) claims that “a work needs only to be interesting”, encapsulating Minimalism’s anti-complexity stance. Wollheim (1965) also frames Minimalist work as deconstructed, in contrast to traditional art’s constructive complexity, and Meyer’s (2004) analysis of Minimalism again focuses on simplicity and restraint.

By introducing the heuristic of narrow decision spaces, we can look at minimalist game design as an approach that parallels these principles, limiting interactive complexity and emphasizing a concise, discrete set of available actions.

Procedural Generation

Nealen et al. (2011) identify procedural generation as another approach for supporting minimalist play. They say that, instead of producing variety through manually authored content, designers can define generative rules that recombine a small set of elements into many possible configurations. In this model, the designer’s task shifts from creating individual configurations to designing the rules that determine how these configurations emerge.

This also resonates with Minimalist art practices. Strickland (2000) highlights Minimalism’s process orientation, where artists employ generative systems and algorithmic logic to produce works with minimal subjective intervention. Perreault (1967) remarked that "what is minimal about Minimal Art is the means, not the end", pointing to the austerity and depersonalization of the method rather than the aesthetic outcome. Batchelor (1997) adds that Minimalism prioritizes pragmatic decisions over craftsmanship or intuition. Strickland specifically says that the Minimal art movement is defined by control, as artists submit to the process and adhere to pre-determined, non-subjective rules.

Minimalism’s focus on algorithmic and procedural creation is particularly relevant to minimalist game design. It offers opportunities for game worlds, levels, and interactions to emerge from predefined, generative rules and logic rather than human craftsmanship.

Using procedural generation as a heuristic, we can evaluate how a game's content submits to predefined, algorithmic rules. It offers us an approach through which generative consistency and

systemic decision-making are prioritized over manual content creation.

Abstraction

Juul (2007) describes abstraction as a design tool that simplifies real-world complexities into more focused game environments. He says that by decomposing complex concepts and removing elements of reality, for example, restricting movement to two dimensions, designers can simplify player actions and reduce the size of the design space of a game. Nealen et al. (2011) extend this to audiovisual design: stylized, non-referential visuals and sound can draw attention to core mechanics and lower the perceived complexity of the system.

Myers (2009) goes further and warns that excessive realism risks shifting games toward simulation, where the essential qualities of games like rules, goals, and opposition become obscured by real-world representation. Nealen et al. reflect this idea, saying that minimalist games push in the opposite direction, omitting explicit narratives so that meaning arises from interaction with the system rather than from storytelling.

Abstraction is also reflected in the logic of Minimalist art. Batchelor (1997) describes Minimalism as non-representational and non-referential, “a world made rather than seen”, directing the audience’s attention away from external associations and towards the inherent properties of materials and forms. Strickland (2000) notes that Minimalist works minimize expressiveness, yielding objects with a factual, depersonalized quality. He says that Minimalism dehumanizes art, shifting from the humanist phenomenology of impressionism and expressionism to a more objectivist approach. In Minimalist music, for example, he says that sounds exist for themselves rather than as expressions of emotion.

This reduction toward a depersonalized quality in Minimalist art resonates with abstraction in game design, where systems of mechanics and rules take precedence over fictional worlds, narrative storytelling and connection with human values.

By embracing abstraction as a heuristic for minimalist game design, we can evaluate minimalist games' ability to offer players a context in which discovery arises from direct engagement with the system, separate from representational expectations.

Wholeness

Finally, I propose “wholeness” as a heuristic to describe the sense that a minimalist game is experienced as a single, integrated whole rather than a collection of parts. Through interactions that feel immediate and complete, cause and effect are perceived together. Instead of layered systems or sequences of mechanics, minimalist games seek to bring action and reaction as close together as possible.

This idea draws on Svanæs’s (2000) discussion of the complex object metaphor, in which micro-interactions combine into cohesive entities or interaction gestalts. According to him, users of interactive systems do not perceive isolated actions but integrated wholes. In minimalist games, this means designing mechanics whose atomic components combine into interactions that feel unified, cohesive and direct.

Wholeness also challenges common interactive design conventions. Norman (2013) mentions hierarchical goal structures which can include sub-goals, reformulated goals, and sequences. This can fragment the experience. Minimalist design instead focuses on simple objectives built from small, consistent tasks.

Morris (1966) argues that Minimalist works rely on unitary forms that avoid compositional detail, hierarchy, or internal relational complexity. He says that they present a complete experience encountered instantly rather than uncovered gradually. Extending this idea of immediacy, minimalist games can offer an interaction space that can be understood directly rather than explored through hidden layers of depth.

A tension exists between Minimalism’s artistic wholeness and game design practice. According to Zubek (2020), games typically rely on progression through the accumulation of mechanics and rising difficulty to sustain engagement. Wholeness, by contrast, resists the unveiling of deeper layers over time. This tension between wholeness and progression can be a site of innovation for minimalist game design.

The challenge and opportunity here is to use wholeness as a heuristic to evaluate how systems can feel complete from the outset but still support discovery through recombination, variation, and emergent patterns.

Synthesis

Taken together, these two sets of heuristics provide a framework through which we can assess how the experience of play is produced under the constraints of both game design and minimalist design. The heuristics for play help evaluate how players act and find meaning within interactive systems, while the heuristics for minimalism proposed here provide criteria for assessing how those systems operate and are built under constraint.

This frames minimalist game design as a dynamic process in which constraints evolve throughout development, interact with one another, and sometimes generate productive tensions that open new avenues for questions and design possibilities.

My research extends this exploration by applying these heuristics as a lens through which I examine the iterative development of a minimalist game prototype. Using these heuristics as analytical tools, I will examine the design process of a minimalist video game throughout its development, highlighting where tensions arise and where convergence happens between minimalism and game design principles.

Methodology

With this project, I adopted a research-creation methodology at the intersection of my creative practice and academic inquiry. With over five years of experience designing experimental video games, I have consistently emphasized minimalism as both an aesthetic and interactive principle. Building on this background, I went through an autoethnographic game design project, documenting my intentions and design decisions throughout the hands-on creation of a minimalist game.

Following Khaled et al.'s Method for Design Materialization, I systematically archived each stage of the design process using online version control repositories. This approach iteratively materializes the design space of the project (Khaled et al., 2018) and enables designers to make scholarly claims by transforming intuitive acts of designing into observable, traceable records or creation (Khaled & Barr, 2023).

Throughout development, I practiced reflection-in-action (Schön & Bennett, 1997), continuously revisiting my intentions, evaluating the outcomes of designing, and iterating accordingly. To enhance the recoverability of my process documentation, I emphasized chronological clarity through consistent note taking throughout the development of the game.

Iterative Game Development

I developed a single player minimalist puzzle game through iterative prototyping cycles, where each feature was designed, implemented, tested, debugged, and refined before moving to the next. This cyclical approach allowed the design and possibility space of the game to evolve organically, while ensuring that the original minimalist intent stayed coherent between emerging features and the overall evolving design philosophy.

Every time I made a modification that significantly altered the game, from a small visual adjustment to the integration of a new mechanic, I recorded it as a commit in a version-controlled repository. Each commit included:

- A description of the changes made
- The rationale behind design decisions that concerned the change
- Questions and hypotheses that surfaced during the process
- Plans for future iterations related to this change

In addition to commit messages, I also maintained several documents in parallel:

- A design journal containing detailed reflections on design rationales and creative choices
- A questions document compiling all the unresolved or recurring design questions that emerged during development
- A to-do document made up of an evolving list of ideas for changes, additions, or experiments on existing aspects of the game
- A feedback document that contained notes and insights gathered from comments players made when playing the game
- A description document which was a living summary of the game's goals, rules, and general intention

Throughout development, I sometimes shared my game with friends, family and people I worked with, usually allowing players to experiment freely with the game, without instructions. While they were playing, I observed their behaviors and noted verbal and nonverbal reactions. This helped me make observations about usability issues in the game and potential directions for exploring the game's design space.

All these records formed a rich and recoverable corpus of design data that captured the evolution of the creative and technical reasoning behind the game's development.

Process Documentation and Data Collection

The objective of this process of documentation was to maximize traceable information about the design process of the game I was making to enable the later analysis of recurring ideas and design concerns related to minimalist game creation.

When I finished the development of the game, all the written material including commit messages, journal entries, and all the other documents was compiled into *Taguette*, a free and open-source qualitative analysis software. This tool allowed me to tag parts of the text according to recurring themes that emerged organically from the data.

Early in process of tagging, several major themes emerged, such as:

- Maintaining playful engagement and interest within a deliberately simple set of possible actions for the players
- Balancing variety and consistency through procedural generation without adding extra mechanics
- Communicating the game's rules and goals while maintaining an abstract, unified and non-representational aesthetic
- Maximizing readability through feedback and feedforward cues, while maintaining mystery and experimentation in play

Thematic Tagging and Heuristic Mapping

After identifying these recurring themes, I realized how they echoed long-standing intuitions and patterns in my own creative practice, particularly the tacit, practical concerns that have shaped my years of making experimental minimalist games. These concerns surfaced during moments of constraint and problem-solving in the design process and formed a consistent intention across my notes and documentation. As I revisited these materials, I found that these implicit tendencies resonated strongly with ideas I encountered in the literature on game design, interaction design, and Minimalism as an art form.

It was through this process of connecting practice-based insights with theoretical perspectives that the two sets of heuristics described earlier began to crystallize.

Using these two heuristic lists as analytical frameworks, I revisited all of my process documentation, tracing how constraints influence and are influenced by emerging features, design decisions, and moments of tension throughout the design process. I organized each text segment (commit, journal note, feedback insight, or to-do entry) by feature and tagged it with

the heuristics that best captured the concerns expressed in that moment. This process of organization allowed me to see how particular features engaged with specific heuristics and constraints through the development of the game.

Synthesis and Reflection

The result of this tagging process was a comprehensive mapping of heuristic concerns across the entire process of designing and developing the game. For each feature or change, I could now identify where the examination of the act of designing through different heuristics converged to reinforce each other or came into tension, revealing the challenges and decision-making dynamics intrinsic to minimalist game design.

In the following chapter, I present an analysis of these relationships. I reflect on how, through the systematic documentation of design decisions and the analysis of areas of convergence and tension through different heuristics emerged between the guiding principles of minimalism and those of interaction and game design.

Process, Analysis and Results

In this chapter, I explore how the design of key game features and the process of navigating a game's design space are examined through the interplay of heuristics for minimalism and heuristics for playfulness.

To ground this discussion, I begin by introducing the concept, mechanics, rules, and goals of the game *Toggle*, which serves as the primary case study throughout this reflection. Following this introduction, I reflect on my creative process and highlight the key moments where the minimalist intention both connected and came into tension with the desire to create something interactable and playful. These moments of connection and tension are particularly important, as they marked points of critical reflection and influenced the direction of the work.

What follows is the story of the creation of a game, shaped by areas of convergence as well as competing intentions between minimalism and playfulness and the design decisions they inspired.

Game Concept and Core Mechanics

The game in question, *Toggle*, originated from a simple idea. In one of my earliest design journal entries, I noted:

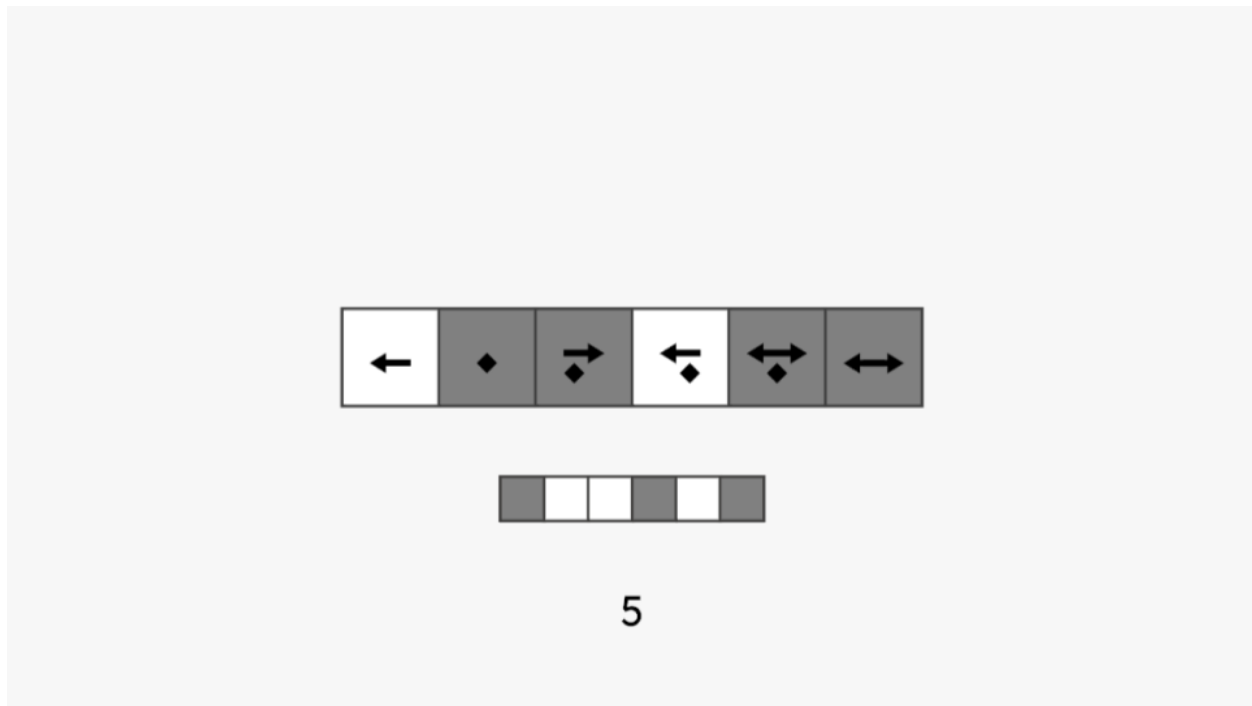
“Today we start prototyping the game's concept! The idea is to present a bunch of squares to the player on a linear one-dimensional array. When clicked, these squares will toggle a state change in themselves and/or other squares in the array.”

(Journal 2024-09-02.1)

As quoted, *Toggle* is a single-player puzzle game in which the player is presented with a linearly arranged sequence of adjacent squares. Each square can exist in one of two states: toggled (black) or untoggled (white). Clicking on a square causes it or other squares in the array to change state. The player's objective is to match the states of these interactive squares to a

predefined "goal array", representing the correct configuration. This must be achieved within a limited number of clicks.

At the time of writing this thesis, Toggle is available for play at <https://majeleger.itch.io/toggle>.



(Figure 1, *Toggle*, commit #41ff233)

Minimal Playable System

The conceptual foundation for *Toggle* is inspired by Dag Svanaes's exploration of interactivity in his 2000 book *Understanding Interactivity*. In this book, Svanaes discusses experiments involving extremely simple interactive systems, such as a single white square that turns black when touched. In more complex iterations, users were presented with arrays of squares where clicking one could affect others (Svanaes, 2000). These experiments were not intentionally games. The engagement was purely exploratory, with no explicit reason to click any specific square. Instead, they served to investigate user interpretations of cause and effect in interactive systems.

I began the work of redesigning and transforming this minimal interactive system into a game. According to Upton, that meant creating a context where players can make meaningful choices. To do that, designers need to add elements like achievable and desirable outcomes, as well as clues to predict and understand the results of a player's action (Upton, 2015).

Expanding the Design Space

I quickly developed the desire to expand on the concept in order to explore more playful possibilities of interaction with these squares. I began to consider how the game could evolve while remaining true to its minimalist core. For instance:

“Should the solution include the possibility of clicking the same square more than once?”
(Journal 2024-09-04.3)

And

“The arrows seem to indicate that the "toggling" will propagate, it does not. Maybe there's a way to experiment with "toggling propagation" at some point in the future?”
(Commits 2024-09-02.2)

Such questions marked the beginning of a broader investigation into the design space of *Toggle*. This exploration continued throughout development and repeatedly made me reflect on the areas of tension and overlap between the two sets of heuristics described earlier: on one hand, Upton's heuristics for playfulness, and on the other, the heuristics I proposed to examine the evolution of the minimalist intention of a game.

Heuristics for Play and Minimalism

Rather than re-explaining them entirely, I summarize these heuristics here to highlight the specific dimensions of play they highlight and that became relevant during the design process:

- **Choice:** games must offer players clear opportunities for action, framed by affordances and constraints. Choices are conscious decisions among possible actions, discoverable

through interaction and repeated across the gameplay loop.

- **Variety:** the set of available choices is not always the same and these choices lead to different outcomes depending on the game's context. Variety prevents repetition and ensures actions remain engaging.
- **Consequence:** every action produces a tangible outcome and a new game state communicated through feedback. This provides weight and purpose to player decisions.
- **Predictability:** players are able to anticipate the results of their actions through signifiers, feedforward, mappings, and consistency. This allows them to build and refine reliable mental models of the system.
- **Uncertainty:** not all outcomes are predetermined. Uncertainty creates tension, challenge, and motivation, requiring players to experiment, fail, and learn through the exploration of the game's possible states.
- **Satisfaction:** some game states are more desirable than others, allowing players to value certain outcomes over others. These rewarding states provide direction and their attainment creates a sense of completion and payoff.

While these heuristics framed my thinking about interactivity and playfulness throughout development, I also worked with another set of principles. As described earlier, these are the additional "minimalist heuristics" I kept in mind in the process of designing *Toggle*:

- **Narrow decision spaces:** the game limits player choices to a small, distinct set of possible actions. This creates focus on the core loop and a small set of interactions, without filler or illusory options. Complexity then comes from the arrangement of simple, discrete rules rather than from a list of expansive choices and complex systems.
- **Procedural generation:** instead of hand-crafted content, the game is built from algorithmic processes that follow simple, repeated rules. This emphasizes system-driven creation over authorial expression, leading to restrained, modular, depersonalized

structures.

- **Abstraction:** the game rejects real-world references, narrative, and representational elements to focus on direct interaction, leading to a clear and geometric aesthetic. Complex concepts are broken down into minimal mechanics, relying on abstract visual clues and direct connection between interactive elements, avoiding decoration or fiction.
- **Wholeness:** interfaces and interactions are designed to feel immediate and complete, forming unified gestalts where cause and effect are perceived together directly. Rather than layered and hidden systems, the game emphasizes atomic actions that stand on their own.

Together, these two sets of heuristics structured the central design question of *Toggle*: how to create a system that is small but engaging, where narrow choice sets, clear consequences, procedurally generated variety, abstract presentation, and actions that feel whole combine in a way that predictability and uncertainty are kept in balance, and that it's still possible to offer simple but satisfying outcomes.

In the following section, I examine how specific features of *Toggle* were implemented within this framework. Each feature is examined through a particular negotiation between the heuristics for play and minimalism, revealing both the areas of alignment and the tensions that shaped the game's development.

Iterative Feature Development

Clickable Squares in an Array

The first feature I developed for *Toggle* was an array of clickable squares, an attempt to embody, as I wrote in my journal's first entry:

“the simplest form of interaction: selecting and activating something”.

(Description .1)

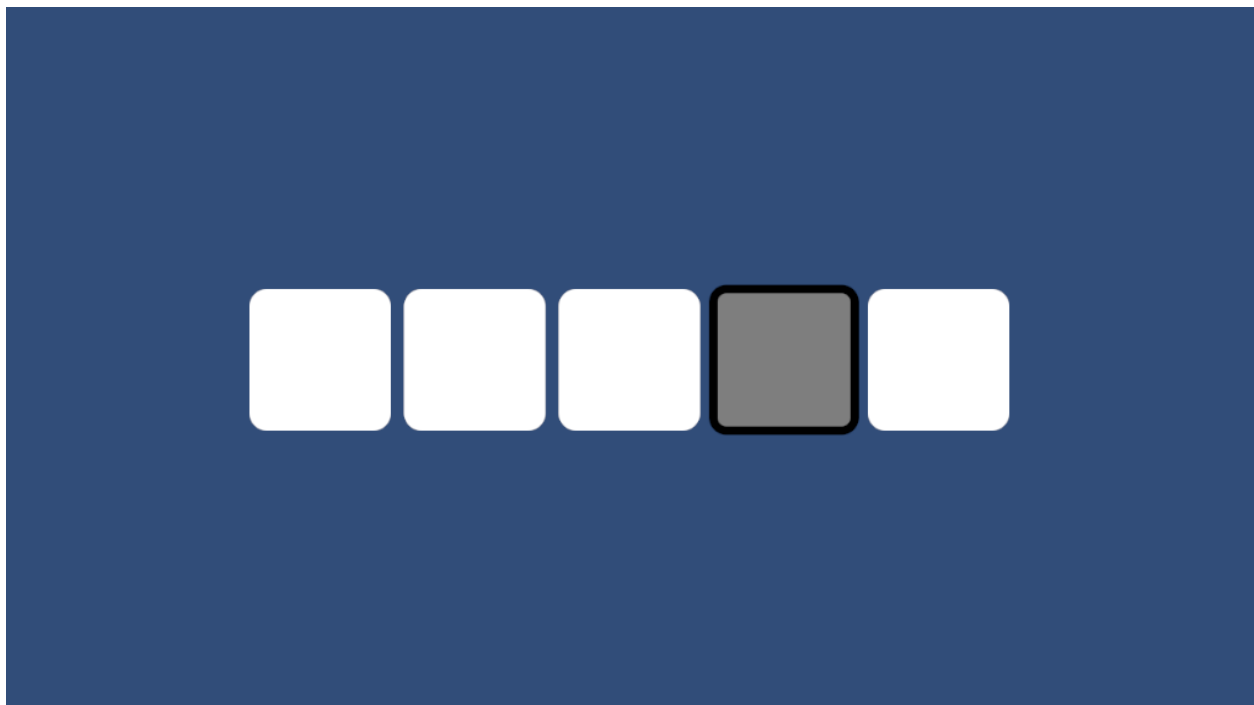
In this array, each square can exist in one of two states, represented by contrasting colors. This extremely simple system provided a foundation for a high degree of discretization, following the heuristic of narrow decision spaces while leaving open the question of how interactions with these squares could create meaningful play.

At the prototyping stage, the concern was not about creating solvable puzzles but about testing whether the idea of clicking squares as the core mechanic of a game could sustain interest. To do this, I quickly and easily produced a level generation system: the number of squares in the array varied randomly between three and six, and each square's initial state was randomized as well.

Interactivity required not only the presence of clickable elements but also the perception that these buttons were clickable. Hover and click effects were implemented almost instinctively:

“There is some feedback on hovering and on clicking the square. It seemed intuitive to implement both in terms of basic interactivity”.

(Commits 2024-09-02.6)



(Figure 2, *Toggle*, commit #85ce024)

The clickable squares feature is a great first example of the negotiation at the core of *Toggle*'s design. On one hand, it was designed to rely on the minimalist heuristics of narrow decision spaces and procedural generation to present players with minimal generative components for interaction. On the other hand, visual clues for these abstract affordances had to be embedded in these components from scratch. This raised a tension between abstraction and the need for signifiers to enhance predictability and the need for feedback to enhance consequence. Although the clicking mechanic created an immediate sense of interaction, it was also obvious that the minimal abstract feedback wasn't enough.

Squares Toggling Other Squares

The next step in development expanded the mechanic from simple clickable squares to ones that could also affect their own state and their neighbors'. As described in one of my early notes:

“When clicked, [...] squares will toggle a state change in themselves and/or other squares in the array.”

(Journal 2024-09-02.1)

This move from self-contained, insignificant clicks to targeted interactions highlighted the importance or consequence. Each click now carried effects on the game's next discrete state.

I integrated a set of targeting schemes to define how squares would interact: toggle self, toggle left, toggle right, or a combination of two or three of these targets. Arrows displayed on the squares indicated their toggle targets, providing minimal visual signifiers. A downward arrow signaled a self-toggle, while side arrows pointed toward neighbor squares. When multiple targets were involved, multiple arrows were combined. These clues attempted to maintain predictability in the system by making outcomes legible, even within the abstraction of a geometric aesthetic.

As with the initial clickable array, procedural generation became a tool for creating variety. Target schemes were assigned randomly when a level was initialized, producing random interaction patterns across the array of squares. In one journal entry, I justified this decision pragmatically:

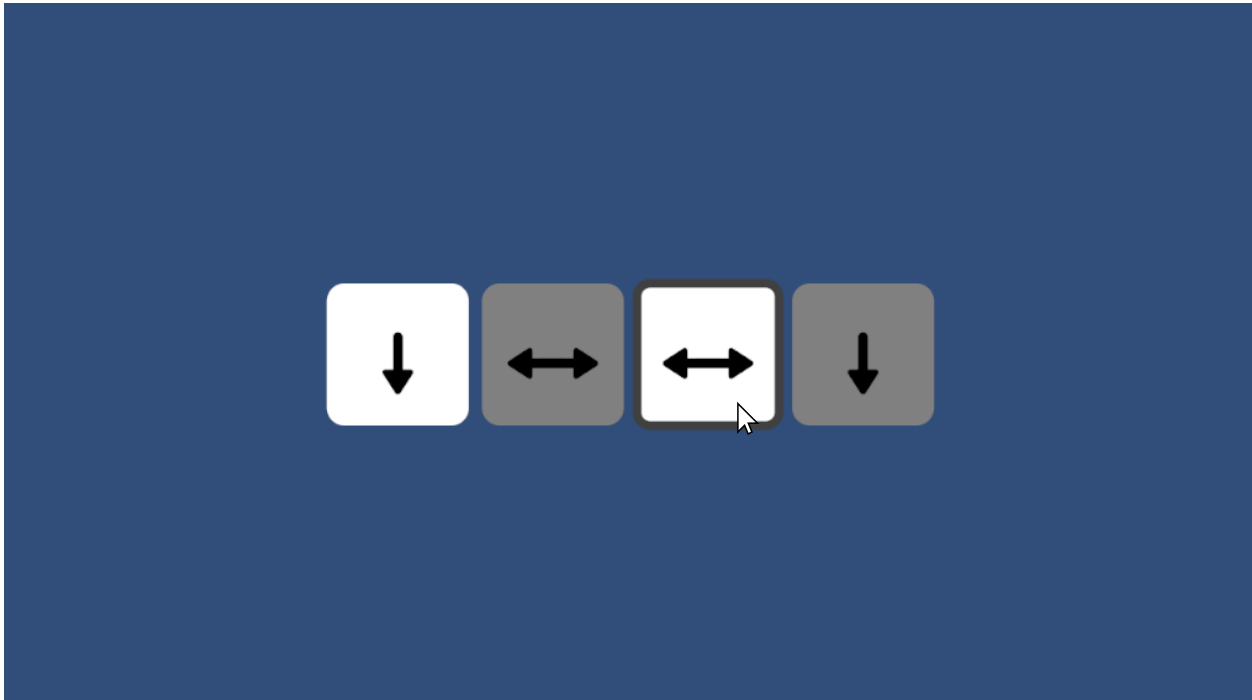
“I'm thinking of generating those level[s] procedurally first, as it'll make it way easier to iterate on different layouts”

(Journal 2024-09-02.5)

Despite the new mechanic, early doubts soon emerged within me about its ability to engage players. In my journal, I admitted:

“Little bit of doubt about the core mechanic (toggling other squares). It's not exactly fun by itself, without requiring a goal to be interesting to play with.”

(Commits 2024-09-02.3)



(Figure 3, *Toggle*, commit #0186e50)

The toggling mechanic created consequence but, without external objectives, it offered little satisfaction. This reflection illustrates how minimalism supports abstract systems that are legible through abstraction and possible to produce through procedural generation, but that risk not being inherently motivating.

Goal Arrangement of Squares

As stated above, without directing the players toward a goal, the system was abstractly interesting but not compelling enough to motivate a gameplay loop. To address this, I introduced a second array of squares, a smaller, fixed, and uninteractable array that displayed a stateful configuration of squares for the player to match. The new feature completed the gameplay loop by giving a purpose to players for clicking squares.

Just as the arrangement of toggling schemes had been generated procedurally, I did the same with goals. The goal of a level was initialized by faking clicks on a random sequence of squares, producing a solvable target arrangement. This procedural approach again provided variety, ensuring each level offered a different puzzle despite the narrow space of possible states. One of my journal reflections captured an initial uncertainty about this absolute randomness:

“This is done because I have no idea of heuristics that indicate whether some aspects of a level’s configuration (or its goal) make it more interesting than another.”

(Commits 2024-09-04.9)

For now, total randomness acted as a placeholder for more deliberate design decisions.

With the addition of a goal, *Toggle* gained satisfaction. Actions no longer only shifted the colors of squares, they moved the system toward or away from a desirable and recognizable outcome. In a later note, I recognized this, while also admitting doubts about the sustainability of the engagement it could generate:

“The addition of a goal makes the core mechanic a little bit more interesting. But something is clearly missing [...] the lack of surprise or discovery or progression within the core mechanisms of the game, maybe.”

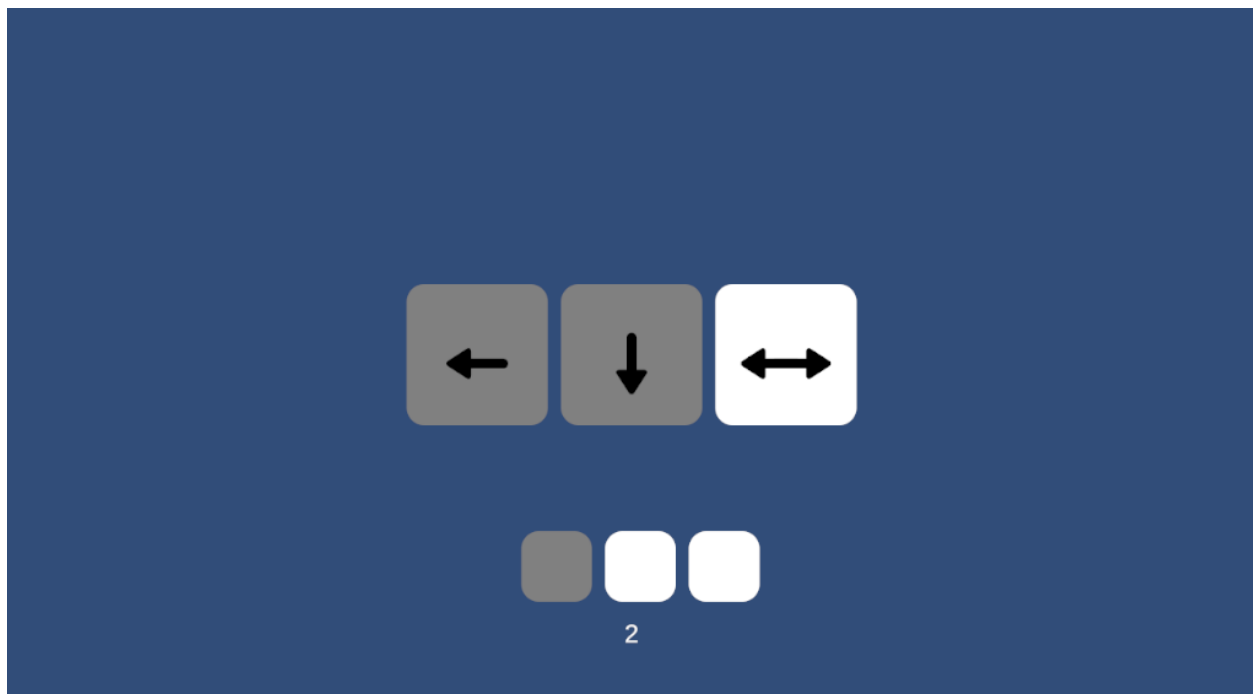
(Journal 2024-09-04.2)

While a feeling of satisfaction was introduced, its depth remained limited by the absence of broader structures such as progression through different mechanics.

Minimalism also made communication difficult. The mapping between the top interactive squares and the bottom goal squares was not immediately apparent to players. One commented that the goal array might look like a progress bar, and many others missed the connection between the two arrays, leading me to write in my feedback notes that:

“The goal of a level (to match the bottom array's colors) is not immediately understood, The bottom array of rectangles is not easily interpreted as the objective of the level.”

(Feedback 2024-12-14.3)



(Figure 4, *Toggle*, commit #38ea59d)

The goal array represented success in adding both variety and satisfaction. On the other hand, it demonstrated the fragility of abstract signifiers and mappings, showing how abstraction can confuse meaning and predictability.

No Trivial Solutions

Procedural generation quickly became the backbone of *Toggle*'s level design. It also produced unsatisfying cases. Some random solutions required only a single click, while others boiled down to clicking each square one after the other. In a journal reflection, I noted:

“already implemented some initial heuristics like: the solution of a level shouldn’t include only one click and the solution shouldn’t be to click all the squares once. [...] These heuristics follow the idea that playful interaction shouldn't be trivial/totally predictable”

(Journal 2024-09-04.5)

Both of these outcomes made interaction trivial. The player faced no real uncertainty, just a sequence of actions that removed any interest in engaging with the puzzle in front of them.

The procedural generation system worked with a small number of squares, binary states, and a limited number of toggling targeting schemes. Within these narrow decision spaces, the challenge was to keep interaction meaningful without going down into absolute predictability. Filtering out single-click and all-click solutions was a way of reintroducing uncertainty, ensuring that the solution of a level could not be immediately predicted without engaging with it.

This decision pointed to a pattern in *Toggle*'s development: procedural systems needed not just randomness but also additional rules to guide the results in the right direction. At this point, it wasn't clear what exactly the level generation algorithm needed to consistently produce engaging levels, but these “negative heuristics” (rules about what not to allow in a level) gave the first shape to the level design space.

Changing the Shape of the Goal Array

To strengthen the visual link between the interactive array and its corresponding goal pattern, I adjusted the goal squares to match the width of the clickable ones. This change flattened their shape, transforming them into rectangles, so they could remain visually distinct while still maintaining a spatial and proportional connection to the interactive array, as noted:

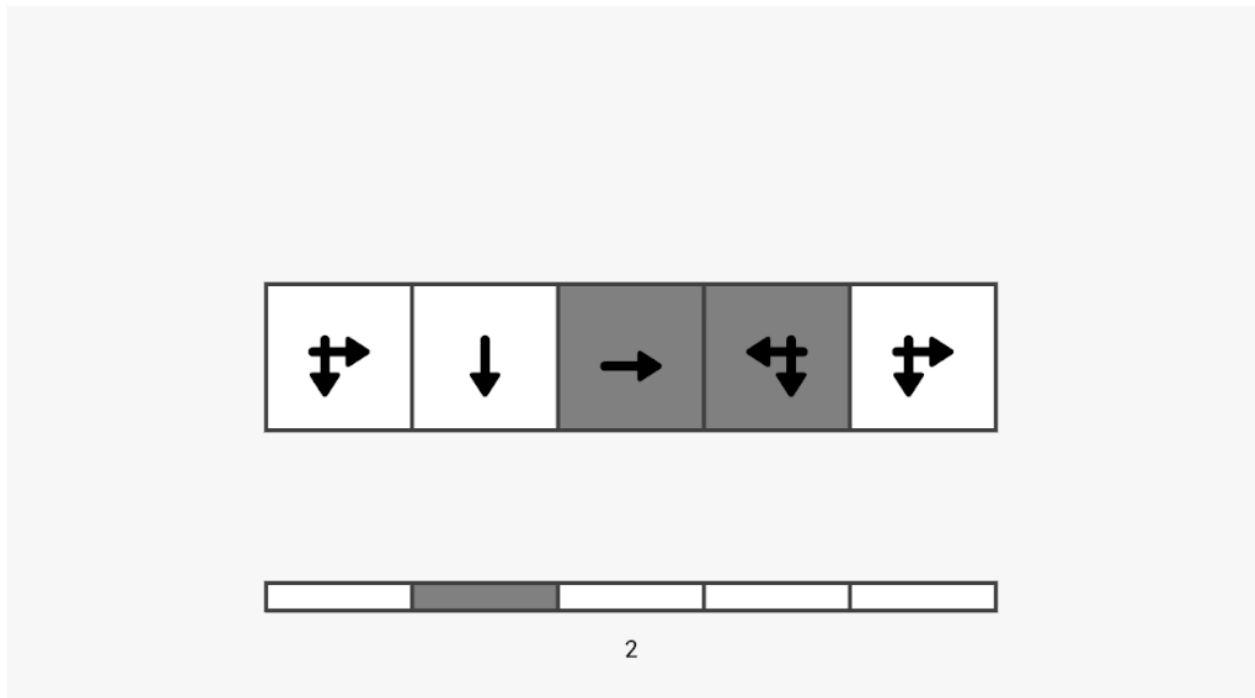
“Made the goal squares as wide as the interactive squares. This is an attempt to fix the weird/unintuitive perceptual mappings between the two. They still needed to look different so I made the goal ‘squares’ a lot flatter.”

(Commits 2024-09-05.4.3)

The goal in this change was to improve predictability, ensuring players could intuitively map and read the correspondence between the squares of both arrays as they were now perfectly vertically aligned. However, subsequent feedback revealed persistent ambiguity:

“Array of solution rectangles might look like a progress bar.”

(Feedback 2024-10-11.9)



(Figure 5, *Toggle*, commit #a8e3268)

The flatter “goal” squares introduced a clearer spatial mapping but also introduced new interpretive issues in some players, showing that no solution was perfect. The change highlighted the tension between visual abstraction and predictability.

Completing and Generating New Levels

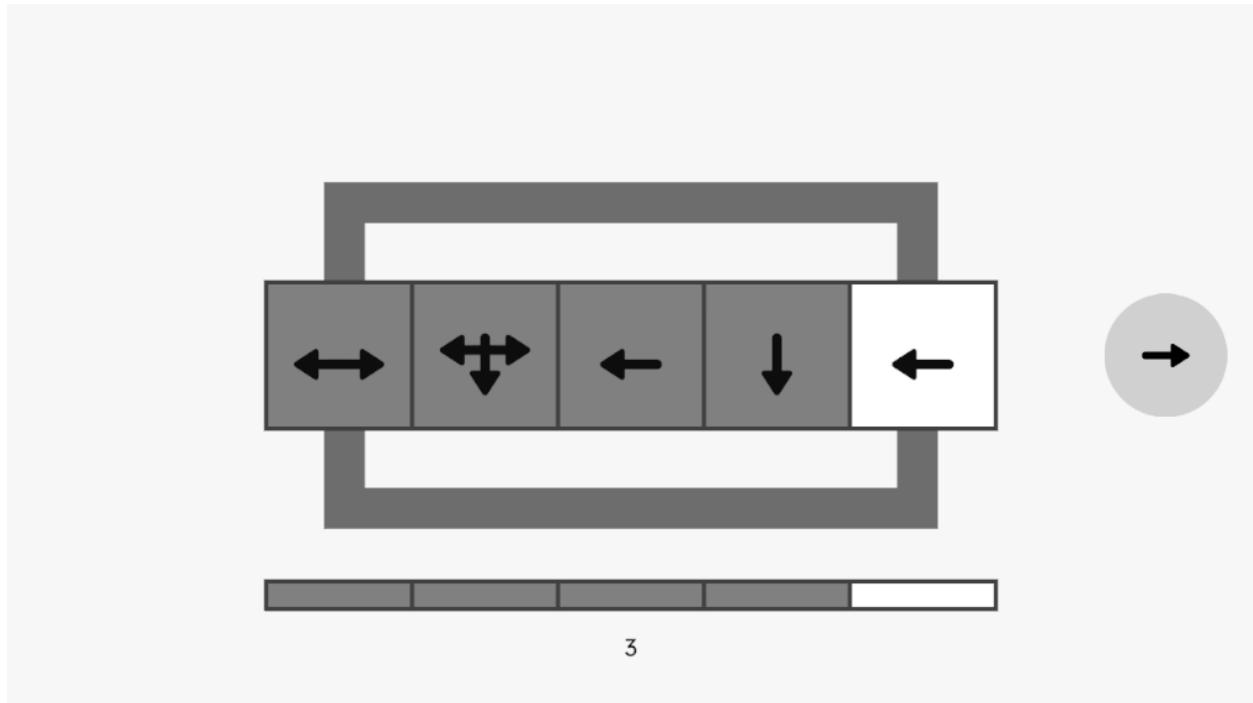
When the player successfully matched the goal arrangement, the game needed a clear signal that the puzzle was complete. After integrating a quick animation that played when the level is complete, I noted in a commit:

“Level completion feedback is half satisfactory for now [...], there’s a button to generate a new level when done with the current one and also the game stops you from being able to click more squares when you complete a level.”

(Commits 2024-09-05.2)

A short animation and removing the interactivity at the end of the level provided closure, preventing the player from continuing to play with the completed configuration. This scheduled limitation on the player’s decision space reinforced the consequence heuristic, showing that each click could eventually lead to a final game state, and created satisfaction by offering a clear payoff. Because of the system’s strong visual abstraction, the end state of a level had to be communicated with minimal means. Rather than elaborate effects or rewards, the solution I chose was to lean on the game’s geometric aesthetic for this animation.

To keep players engaged in the gameplay loop, they could generate a new random level after completing the current one by clicking a “next level” button that appeared on the right side of the screen. The button’s only visual was only a right-pointing arrow, abstractly suggesting forward movement and continued progress. It remained inactive until the level was solved, reinforcing a sense of closure and signaling that it was time to move on when it appeared. Making this button clickable also produced a satisfying state: signaling to the player that there was now more levels to come.



(Figure 6, *Toggle*, commit #9670c92)

Each new level was generated algorithmically, producing infinite variations while following a handful of rules and constraints. This once again supported variety through procedural generation, allowing the player to be presented with new levels without manual level design.

Undo, redo and reset functionalities

By introducing undo, redo, and reset functionalities, I brought a layer of navigability to each level. The three buttons allowed players to backtrack or re-explore their sequence of moves, or to simply start over. I disabled these features once a level was completed in order to strengthen the sense of closure when finishing a level.

“Implementing undo and redo features this early in development is kind of bothersome [...] but I do feel like undoing and redoing is important in these types of puzzles [...] they allow players to navigate puzzle-solving trajectories and histories.”

(Journal 2024-09-06.1)

After their implementation, these functions prompted reflection:

“Undo and redo actions still have to make sense interactively, even though their effects are a bit disjoint from the game’s interface compared to the ‘regular’ actions they revert back.”

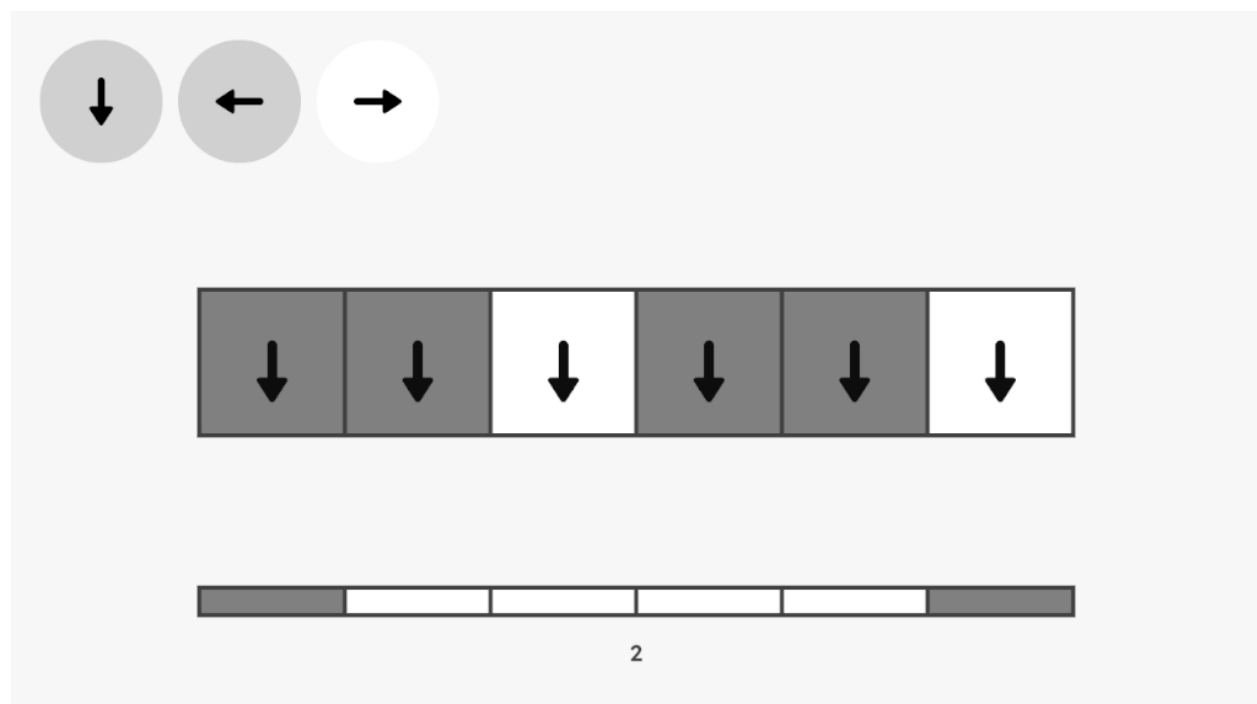
(Journal 2024-09-06.2)

Regular actions (clicking a square and seeing its toggle effects) were direct and tangible, enhancing wholeness. Undo and redo, however, were indirect actions. Their effects were on the history of the level rather than the level’s objects. This distinction posed a design challenge: players had to perceive the causal link between pressing a button and the game state shifting backward or forward in time. As I noted, making sure that the interface signified this clearly became a concern:

“Undoing or redoing a move feels weird when you can’t actually see the ‘move’ go in reverse... Now it just seems like another move is being made.”

(Commits 2024-09-06.9)

I intended to make the game and its interface as static and whole as possible, so the absence of movement or animation was at least partially responsible for this issue. Without visual cues of clicks going in reverse, undoing felt indistinguishable from performing a new action.

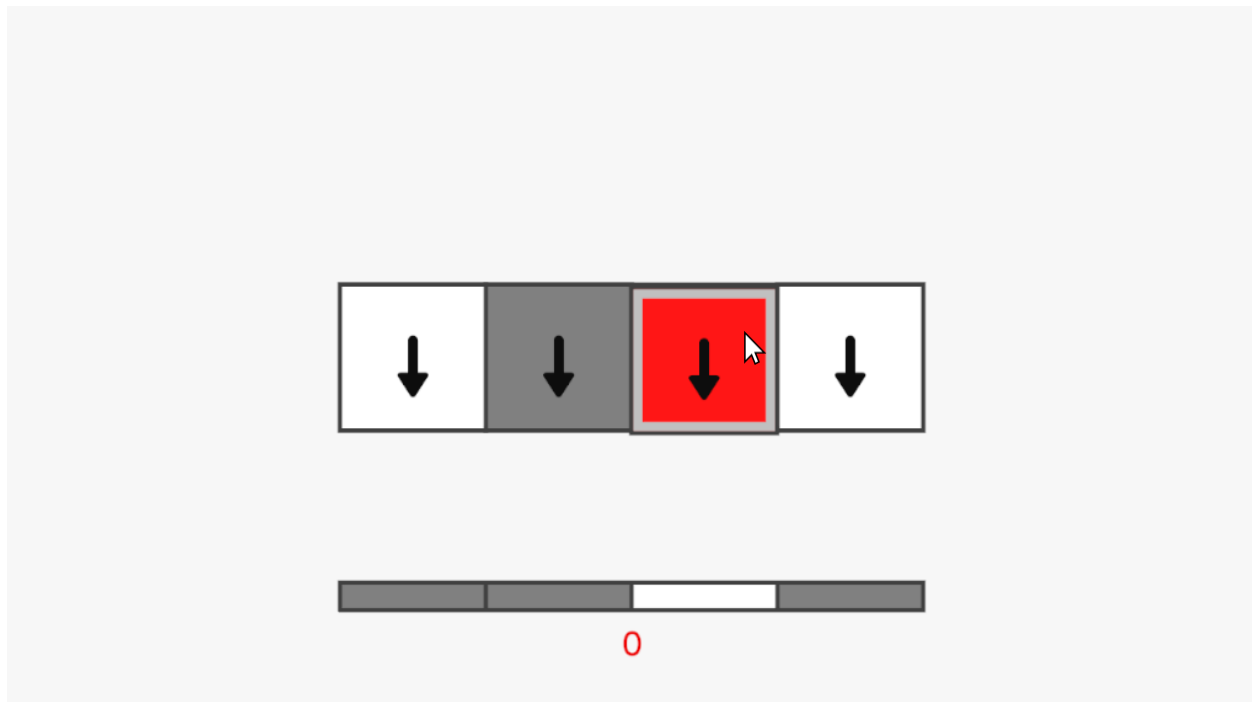


(Figure 7, *Toggle*, commit #dadcbbb)

The process of making these functionalities understandable by the player provided an important reflection on the tension in communicating indirect actions within a minimal interface. While direct interactions like clicking a square and seeing an immediate effect could be conveyed cleanly through simple, abstract feedback, indirect actions like undo, redo and reset required a different kind of legibility.

Clicks Restriction

Next, I introduced a limit to the number of clicks available per level. Each level displayed a click counter that counted down with every press on a square, enhancing the feeling of consequence of every click. Once the counter reached zero, the player could no longer toggle additional squares and attempts at doing so triggered a brief red pulse and shake on the clicks counter, signaling a hard boundary within the game's system.



(Figure 8, *Toggle*, commit #041ca67)

Player reactions revealed both issues of predictability and potential for satisfying feedback. As a very abstract visual element of the interface, the click counter was sometimes mistaken for a

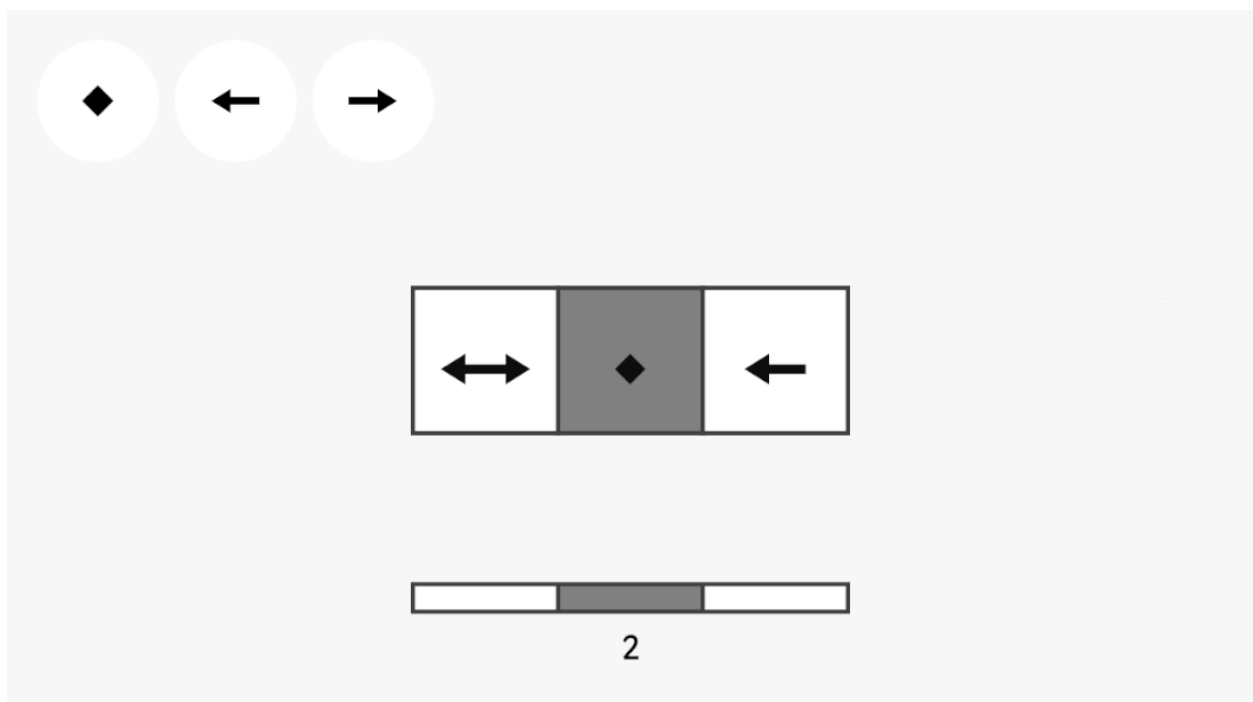
level number or overlooked altogether, suggesting the need for a clearer and more prominent presentation, but players who understood its function described the red-shaking response as satisfying, appreciating the clear boundary and consequence it introduced.

Self-Target Indicator Change

I changed the self-target icon from a downward arrow to a diamond shape, hoping to prevent confusion. Early playtests showed that the down arrow was easily misread, especially within the game's abstract visual language.

“I should find a better way to indicate a self target than a ‘down’ arrow.”

(Commits 2024-09-04.8)



(Figure 9, *Toggle*, commit #d90dad8)

The change served both predictability and abstraction: the intention of the diamond, centered and symmetrical, was to convey self-reference without implying a direction.

No Pre-Solved Levels

An issue in the procedural generation system was that some levels appeared already solved, offering no interaction or challenge. I addressed this by adding a rule to exclude levels generated with a goal configuration that was identical to the initial configuration of the clickable array.

“Not generating already solved levels anymore. This is [one of] the first and very obvious heuristic to implement in order to generate ‘interesting’ levels.”

(Commits 2024-09-19.6-7)

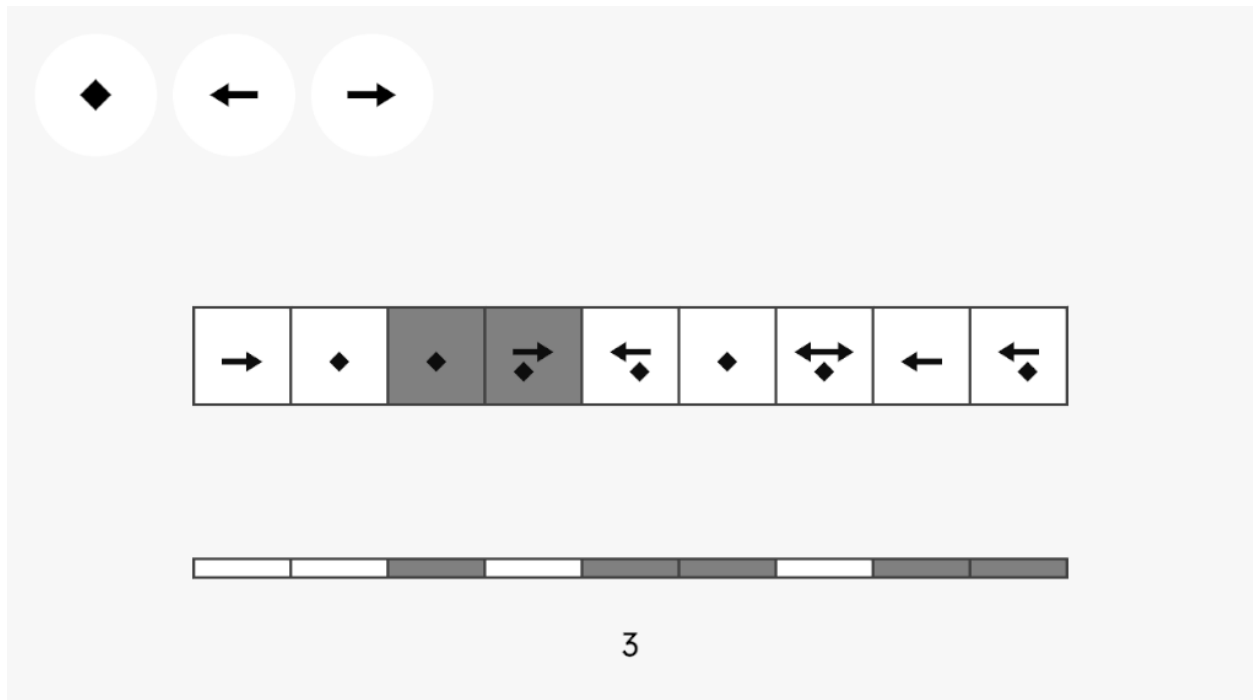
This change reinforced the heuristic of choice and satisfaction, ensuring every level presented at least one meaningful action to take in the journey toward its goal. It also reflected a growing concern for consistency in the game’s procedural generation logic.

Longer arrays

To increase variety and broaden the expressive range of possible puzzles, I expanded the array size to include up to nine squares. This required pulling back the camera and slightly reducing the overall scale of the interface to make the longer levels fit nicely within the bounds of the screen and preserve wholeness.

“The intention is to generate more diversity in randomly generated levels. I felt like a square range of 3 to 6 wasn’t allowing levels diverse enough to give a concrete idea of the range of possible puzzles.”

(Commits 2024-09-25.9)



(Figure 10, *Toggle*, commit #6a44792)

This change also made me reflect on the level size limits. While it would technically be possible to create levels spanning up to 20 squares or more to create even more variety, doing so would introduce the need for scrolling or zooming, interactions that would counter the heuristic of wholeness, which is maintained by keeping all parts of a level visible and directly accessible without fragmenting the player’s interactive experience.

Prediction Rectangles

There was a lack of feedforward to help players make informed decisions. To address this, I introduced a system of prediction rectangles. When hovering over a square, rectangles appeared beneath the target squares, previewing which ones would toggle, and into which state, if clicked. I reflected:

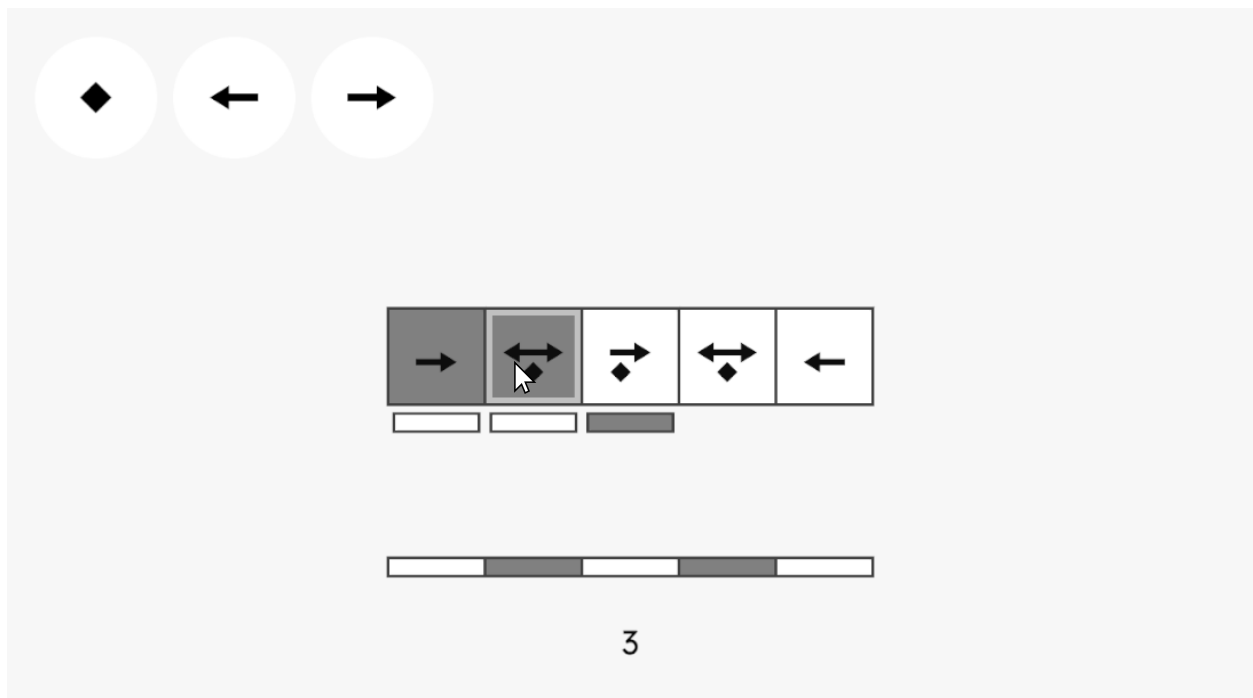
“The rationale behind implementing these minor modifications before working on more substantial pieces (like new mechanics and diving deep into the level generation algorithm itself) is because I want to get the feeling of playing with the squares right before I make the game more complex. I want players to understand what’s going on, first and foremost.”

(Journal 2024-09-25.2-3)

This change emerged from a recurring concern: the interaction felt too opaque. Without visible clues, players were unable to anticipate the outcomes of their actions, making the game feel arbitrary rather than minimalist. The prediction rectangles therefore became a way to add predictability without adding textual explanation, while staying true to the aesthetic of abstraction and minimal visual communication.

“Affordances are hidden, therefore feedback and feedforward and signifiers are doing all the heavy lifting. [...] It's making interaction with the squares richer, in that it supports anticipation, not just trial and error.”

(Commits 2024-09-25.4-5)



(Figure 11, *Toggle*, commit #8b65ce9)

This led to a later consideration about how much information the rectangles should reveal:

“Should the preview rectangles show the whole predicted array, not just the “changed squares”? This would make it easier for players to predict their next move. Do we want to make it easier for players to predict their next move?”

(Questions 2024-10-17.2)

This reflected the start of an ongoing tension between uncertainty and predictability. The prediction rectangles became a tool for negotiating between the two: maintaining challenge and playfulness while giving just enough visual structure for the system to feel learnable.

No Useless Arrows

A change in the level generation algorithm that I made was to remove arrows pointing toward empty sides of the array, in which case a square's directional toggle would target nothing.

“Squares can effectively be ‘useless’ if they only have a left arrow and are positioned at the beginning of the array, or only a right arrow and positioned at the end. I’m wondering if this is an interesting bit of obfuscation? Or will it be unnecessary noise in the player’s reading of a level?”

(Commits 2024-09-03.2)

This change was an application of the heuristic of consequence, in that every visible affordance should lead to a meaningful outcome. By removing these superfluous arrows, levels became easier to read and play, reducing visual noise while narrowing the decision space.

However, as seen in this excerpt from my journal, I also reflected on the potential of visual noise to create playful ambiguity. This reveals a subtle tension between the heuristic of consequence and the heuristic of uncertainty.

Difficulty Through Click Counts

To create a more consistent sense of difficulty, I made the number of clicks required to solve a level to always be one fewer than the number of squares in the array. This replaced the random range where even long arrays could be solved in only a few moves, which felt underwhelming.

“Before, it could go as low as 2 on any level and that felt really underwhelming on longer arrays. I changed it so it’s always 1 fewer than the number of squares in a level [...] The intention is that longer levels are consistently more difficult than smaller ones.”

(Commits 2024-09-26.5)

This change reflects a link between variety and difficulty scaling, ensuring procedural generation yields levels whose challenge is proportional with size. The relationship between array length and complexity remained to be tested, but this simple rule helped maintain consistency across the set of generated puzzles.

Undo, Redo, and Reset Available After Level Completion

I made undo, redo, and reset functions available even after a level was completed, allowing players to replay and re-solve the same puzzle in different ways. This design decision transformed level completion from a single endpoint into an opportunity for experimentation, and encouraged players to explore alternate solutions rather than simply moving on.

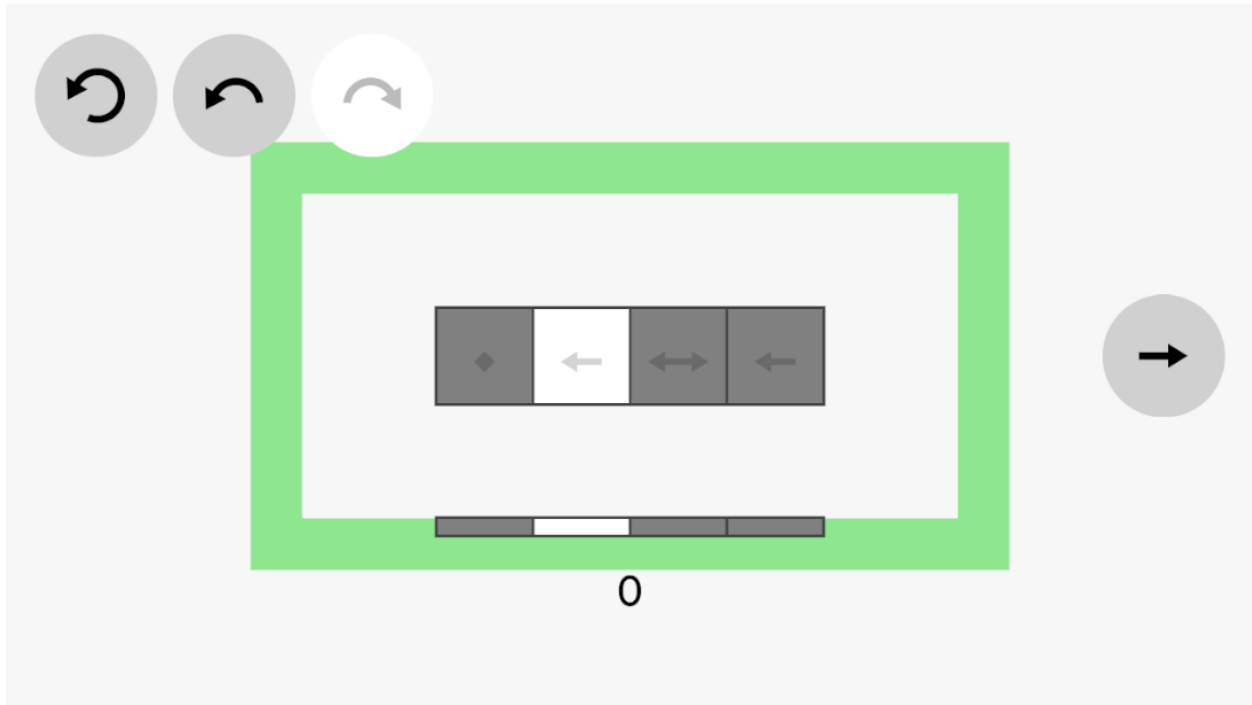
“Undoing and resetting is possible when a level is completed. This opens up the possibility of players completing levels with different amounts of clicks. This way, we can reward them for specific solutions.”

(Commits 2024-09-26.6)

I also introduced a subtle change in the reward mechanism: completing a level with the exact number of clicks specified by the counter would trigger a special colored animation. I justified it in my journal:

“In a world of black and white and grey, green should be appreciated in a special way.”

(Commits 2024-09-26.6)



(Figure 12, *Toggle*, commit #1337c7b)

This change strengthened player choice by adding the possibility of post-completion experimentation, while maintaining a narrow decision space. It also provided more satisfaction through achieving different goals while maintaining a minimalist aesthetic through abstract feedback.

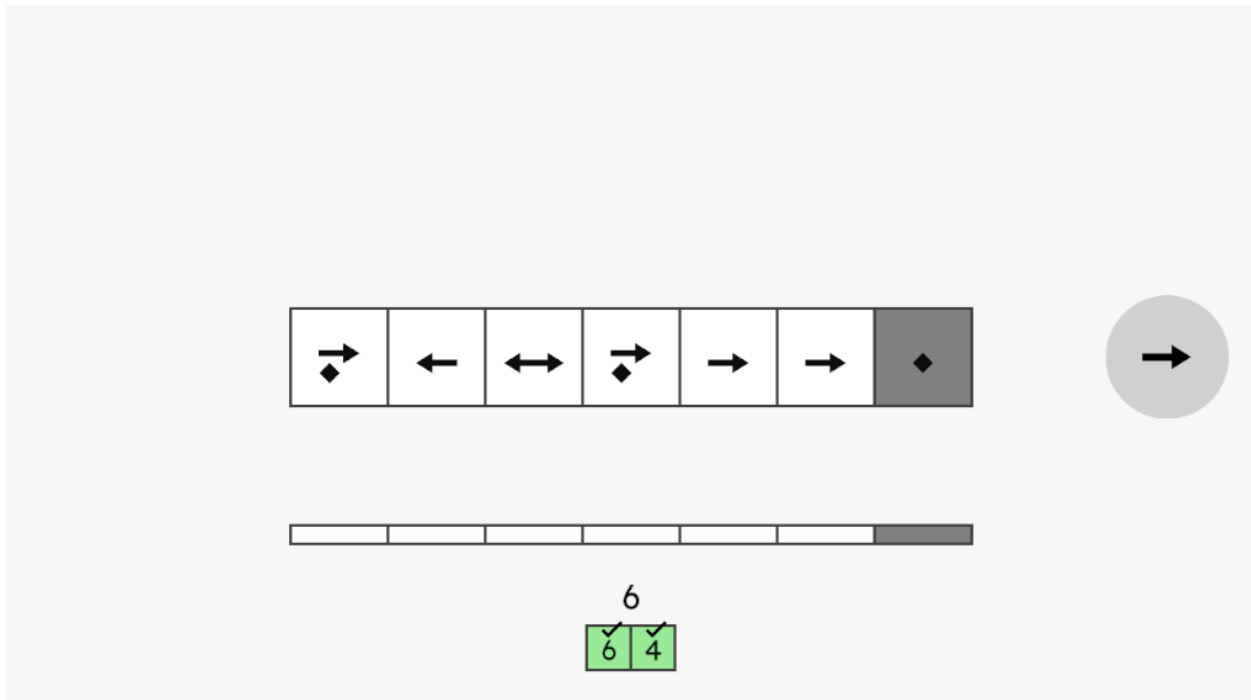
Multi-Solution Mode

By exploring the design space of allowing players to complete levels using alternate solutions, I decided to make the “multi-solution” loop an official feature of the game. Whenever a generated level contained multiple valid ways to reach its goal configuration, the interface displayed multiple solutions markers, each represented by a distinct number in a box above the click counter. Players could see that a level contained more than one solution and were invited to find them all. Levels were then only considered fully complete once every solution had been discovered, each corresponding to a different number of clicks.

“Displayed the number of clicks for every possible solution of a level on the UI. This indicates the broader goal of a level: playing out all possible solutions. It invites players to do so even though they might not have realized multiple solutions were possible.”

This mode created a new loop for the players. Instead of simply getting to the goal once, they had to re-enter the puzzle space and search for alternate paths within the same configuration. In doing so, the mode supported variety and choice while maintaining narrow decision spaces since the player's basic actions remained the same even within this modified gameplay loop.

However, this feature also revealed an important design tension. Some of the generated and suggested solutions were trivial variations of others, leading to moments where uncertainty decreased and repetition set in. This revealed a limitation of procedural generation: too much variety of possible solutions could make different interactions with the same puzzle feel mechanically similar. To address this, I experimented with restricting each square to a single click per level, intending to narrow the range of possible generated configurations. This extra change helped restore a balance between variety and uncertainty in the level generation algorithm's output.



(Figure 13, *Toggle*, commit #f1cfe9e)

When players commented on their preference between the two modes, responses were mixed. From these reactions, I summarized :

“Multi-solution mode is mostly frustrating [...] It feels more like busywork. [But it also] creates some tension and a teasing sort of engagement [...] Squares being ‘spent’ in multi-solution mode feels good and captures the puzzly feeling.”

(Feedback 2024-11-08.4-7-13)

“The two modes (multi-solution and single-solution) don't have to be mutually exclusive. The game could progress from single-solution to multi-solution levels (this suggests that multi-solution mode is generally harder than single-solution).”

(Feedback 2024-11-08.1)

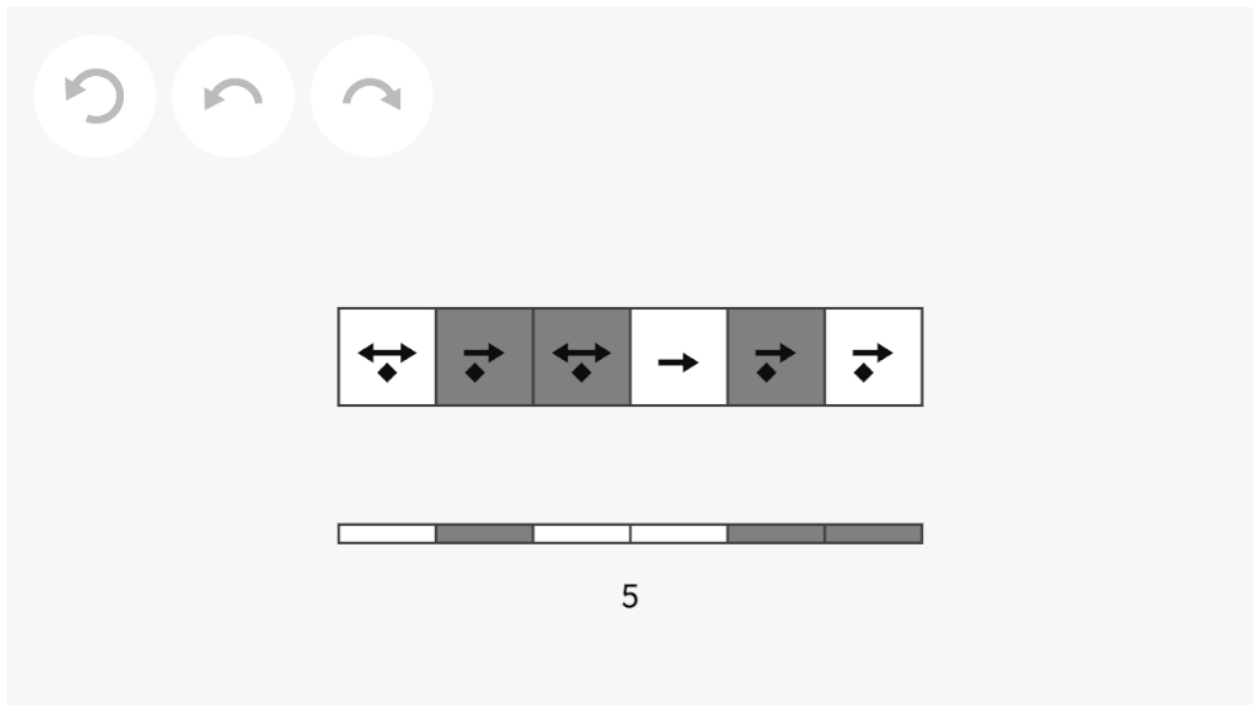
Because neither of the two modes proved to be a clearly superior experience, I made it so both single-solution and multi-solution modes remained available in parallel during most of *Toggle*'s development. This flexibility allowed for future iterations on level design and difficulty progression.

Wrap-around Toggles

The goal of adding this feature was to break the linearity of puzzle solving in *Toggle*. The wrap-around toggles allowed the first and last squares of an array to interact with each other, creating a circular relationship between the game's squares. Arrows that point out and away from the ends of the array (the ones that were previously considered useless) were reintroduced and repurposed to toggle the opposite side's square instead, turning the array into a loop. This was inspired from an observation I made when watching someone play:

“Solutions can be figured out too linearly by going from left to right. Introducing ‘wrap-around toggles’ for the end squares might be a solution.”

(Feedback 2024-10-11.12-13)



(Figure 14, *Toggle*, commit #683a0b6)

This small change had a significant impact on how players approached the puzzles. It introduced uncertainty and variety to the challenge. The game could now generate levels that included this new mechanism and made it so they couldn't apply the clear left-to-right reasoning that felt too predictable before. The player had to consider the array differently, where squares could now influence more than their visually adjacent neighbours. This disrupted predictable patterns and made the narrow decision space into something richer without adding entirely new mechanisms.

From a development standpoint, the change was minimal, but it created new possibilities within the game's system. This also reflects how, in such a reduced design space, richness in gameplay can emerge not from adding new elements, but from reinterpreting existing ones.

Enforcing Number of Solutions

The objective of this feature was to ensure that the structure of each level aligned with the expectations established by its mode. In single-solution mode, I made it so only levels that could be solved in one way were generated, while multi-solution mode only generated levels with multiple possible solutions.

“It felt weird to generate levels that didn’t directly fill the properties of the mode they were in. ‘Multiple-solution mode’ shouldn’t just allow for multiple solutions, but offer only that, and vice versa. Thinking of affordances, it would feel weird if a door afforded opening, but only sometimes.”

(Commits 2024-10-15.13-14)

This change enhanced the consistency and wholeness of the game’s design. By enforcing the relationship at the algorithmic level, each mode became a more coherent experience. The affordances of each mode were now more clearly communicated through both mechanics and win conditions.

However, this added constraint also had the consequence that some configurations of squares made it mathematically impossible to produce multiple valid solutions.

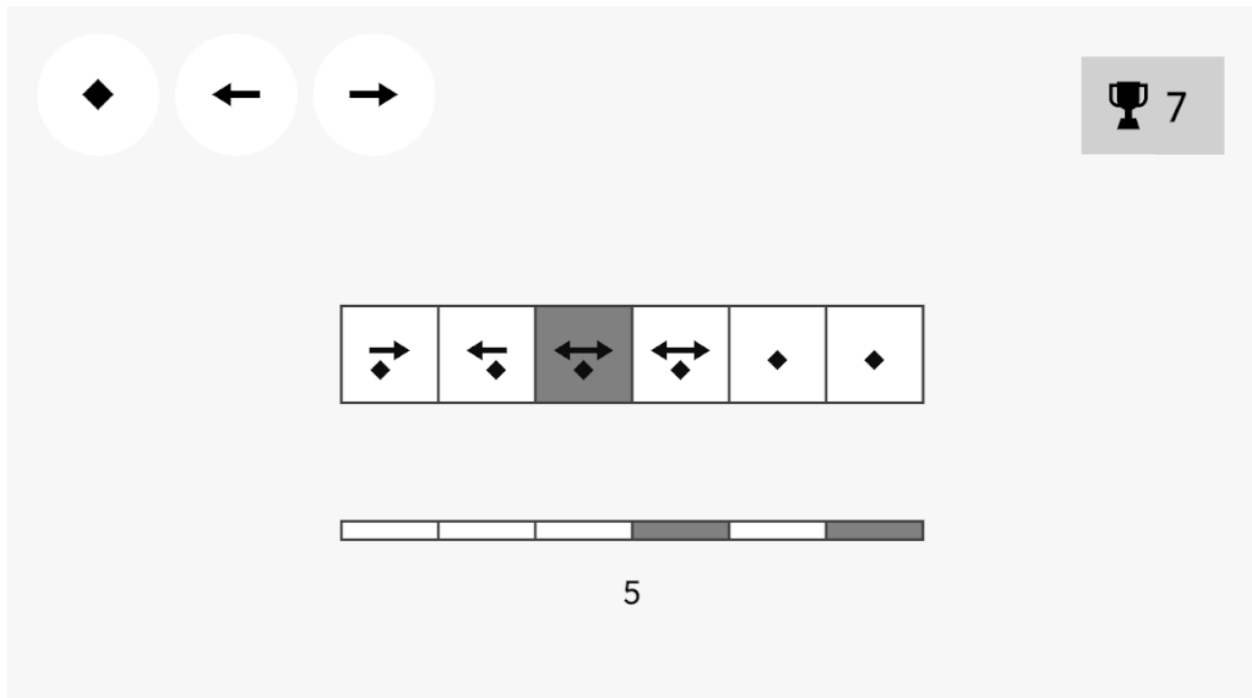
This change demonstrated the balance and tension between process-driven content generation and variety. In other words, constraints that aim to ensure consistency strengthens wholeness, but too much of it can limit emerging variety.

Progression

I introduced a progression system in order to give the game a stronger sense of evolution and discovery over time. Using this system, levels were now generated according to a gradual curve of difficulty and mechanical complexity, rather than appearing as randomized puzzles. The system tracked the number of completed levels, represented by a trophy icon and counter on the UI, and used this value as a progression index that controlled which features and array lengths could appear in subsequent levels.

At that point in the development of *Toggle*, this progression system made it so the clickable array of squares expanded in size by a new square every two completed levels. This linear increase deepened the decision space as the game progresses while maintaining the same narrow set of possible actions. Additionally, at the beginning, players encountered only the most simple arrow schemes (self-targeting and simple left/right targeting) while later levels introduced more complex combinations of these. Similarly, wrap-around toggles were introduced later so

that players first become comfortable with the simpler, bounded array logic before having to think cyclically.



(Figure 15, *Toggle*, commit #06febdf)

This helped maintain a balance between predictability (players knowing things would get gradually more complex) and uncertainty (not knowing which mechanic would appear next).

I was motivated to introduce such a system by the lack of surprise and discovery in earlier iterations of the game:

“Something is clearly missing [...] the lack of surprise or discovery or progression within the core mechanisms of the game, maybe.”

(Journal 2024-09-04.2)

Also, without a structured progression, the gameplay loop felt static. Players quickly saw all possible interactions and either lost interest or became overwhelmed. Early feedback from players confirmed this:

“Someone likes the totally random, not progressive way levels are spat out. Most everyone else disliked that and would enjoy a progressive level generation scheme instead.”

(Feedback 2024-10-11.4)

“[Players] felt overwhelmed beyond the first level (too many squares?) [...] An actual progression in the generation of the level would help a lot. Players learning by themselves might actually be a source of motivation.”

(Feedback 2024-10-11.7-8)

However, this new system also added constraints to the procedural generation process, occasionally making it more difficult to produce valid configurations:

“Also to note that having more restrictions on which arrows to choose from in the generation of a level does hinder the level generation algorithm.”

(Commits 2024-10-31.3)

Overall, the progression system created a more cohesive sense of discovery and flow, transforming the experience from a sequence of disconnected random puzzles into a structured journey through variety, introducing new features and configurations gradually to sustain engagement. It also enhanced predictability, creating a clear rhythm of learning, as well as uncertainty, creating surprise through staggered feature reveals.

Finally, progression also reinforced the heuristic of consequence, linking player actions to concrete, meaningful evolution in the game world. Yet it also introduced a subtle tension with the minimalist heuristic of wholeness, since certain features of the game now remained hidden until a specific point in progression was reached.

Variation in the Solution Found Animation

A small adjustment I made next was to change how the game communicates success. The “solution found” animation was previously triggered every time a level’s goal arrangement was reached. I changed it so it only appeared when the player discovers a new, previously uncompleted solution.

“It was a confusing piece of feedback to be shown the ‘Solved’ animation when you didn’t actually solve anything new.”

(Commits 2024-11-04.4)

This change reinforced consistency in the game’s feedback system and strengthened the heuristic of consequence by making sure to match the visual responses of the game’s interface with the significance of the player’s action. However, player reactions later revealed that this also introduced ambiguity, as some players interpreted the absence of animation as a sign of failure:

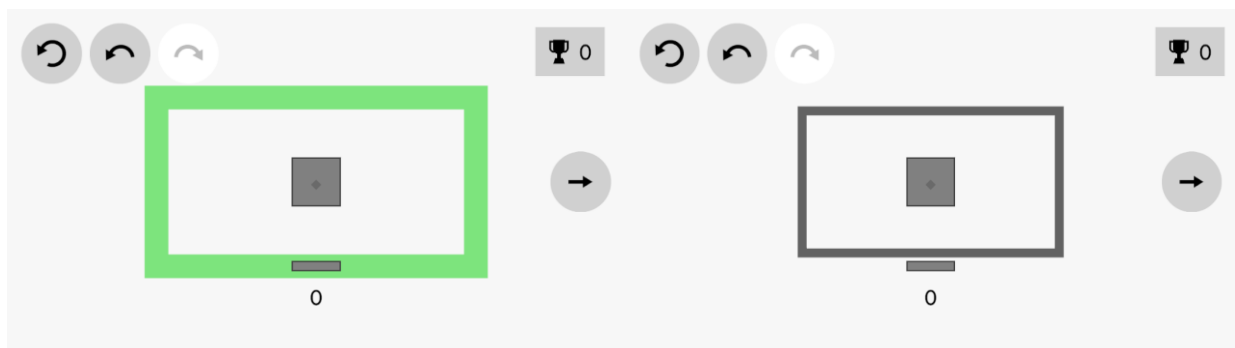
“The fact that the winning animation doesn’t play every time the level is re-solved makes the player doubt if they actually successfully understood the goal and how to reach it.”

(Feedback 2024-12-14.2)

In response, I reintroduced the animation for all completions, but with a visual distinction between first-time and repeated successes.

“[I] made the level completion animation play every time a level is re-solved, with a bit of a difference [...] I made the graphic thinner and less colorful every time it’s animated after the first time.”

(Commits 2024-12-27.8-10)



(Figures 16 and 17, *Toggle*, commit #f32b758)

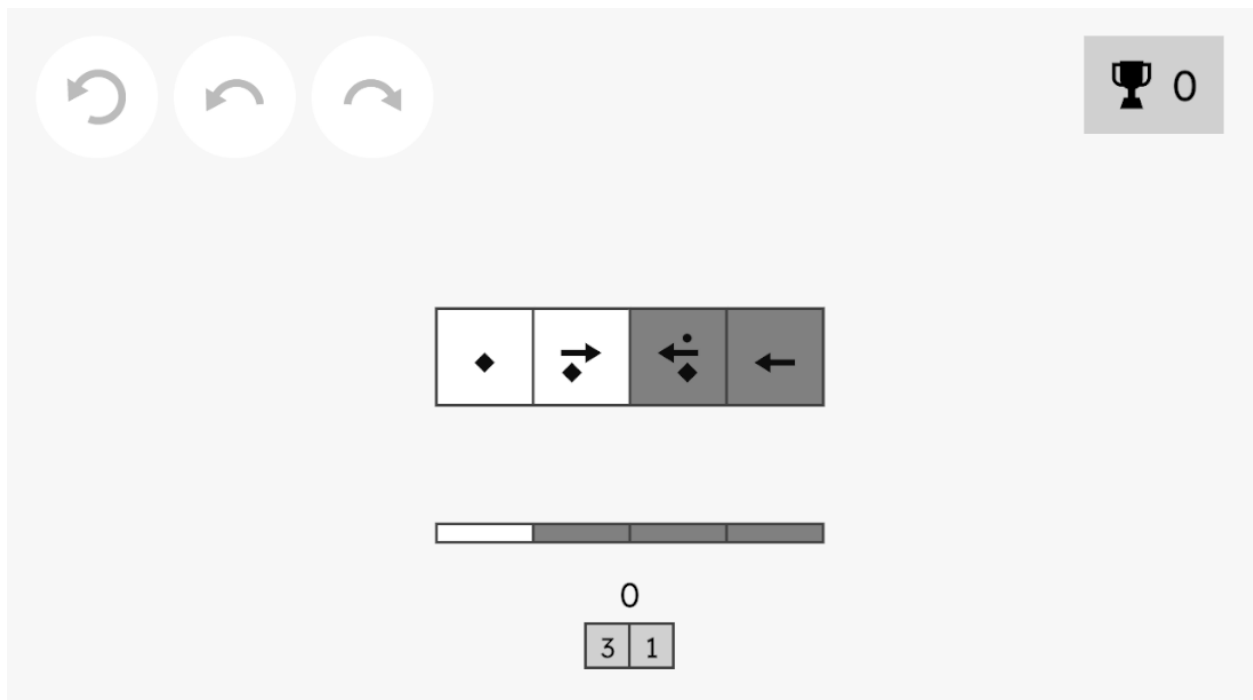
This quick iteration cycle was used to balance consequence and abstraction and ensured that players always received confirmation that their input was correct, while still subtly communicating whether a solution was new or already completed.

Cascading Squares

This feature represented a significant moment in the development of *Toggle*. I introduced a system where toggling one square could trigger a chain reaction: some squares, when toggled, would immediately toggle their own targets as well, creating a web of simultaneous state changes.

“A dot on a square indicates that, when it is targeted and toggled by another clicked square, it will toggle its own target(s) immediately as well.”

(Description .4.4)



(Figure 18, *Toggle*, commit #370f9ad)

This feature originated from early reflections about toggling propagation, a mechanic I first mentioned in my journal in passing as something that might be worth exploring later:

“The arrows seem to indicate that the ‘toggling’ will propagate, it does not. Maybe there's a way to experiment with ‘toggling propagation’ at some point in the future?”

(Commits 2024-09-02.2)

Months later, the idea resurfaced during a reflective period when I was writing about trying to “find the fun” in the project:

“I've been trying to refine the core loop to make the fun ‘pop out’ of it. Maybe I need to do more than ‘refine’ it. Maybe I need to twist it in a different direction instead [...] My first intuition is to brainstorm a list of different mechanics than the basic ones that I have implemented. First thoughts include: having squares toggle other squares in a maybe less intuitive, more interesting manner(s) [like] activating the toggle mechanisms of other squares when toggling them.”

(Journal 2024-09-19.2-3-4)

The cascading toggles became that twist, as a way to evolve the same mechanic without expanding the game’s narrow decision space.

“I feel like this kind of twist would be a very interesting evolution on the game’s core mechanism, as well as being very minimalistic in nature since it doesn’t change or add a new way to interact with the game. Rather, it changes how the game resolves the same mechanic.”

(Journal 2024-11-04.2)

The integration of cascading squares raised many new design questions about difficulty through uncertainty, predictability and feedback. For example, since all cascades happened simultaneously, squares toggled an even number of times appeared unchanged, making cause and effect very difficult to read.

“It works, although there is no animation or feedback yet. I still have no idea how to make this feature look and feel like what it does functionally but I at least wanted to test the functionality, to play with it and figure out if it was interesting or not.”

(Journal 2024-11-04.4)

“It makes the whole game a lot more complex, not just a little, especially without feedback. Is that level of complexity something that we want?”

(Journal 2024-11-04.7)

“Should there be something in or around the prediction rectangle when a cascade is predicted? Clear and precise feedforward? Or some vague/simple indicator that something’s about to happen?”

(Journal 2024-11-04.12)

Despite these complex questions, this feature proved to be a great example of expanding the design space of a game minimalistically. It incorporated a new way to think about interacting with the game’s system through a small change in the action-reaction scheme of the game.

“The code was pretty much already all there; I just needed to add a line or two to re-call the ‘click’ function on a toggled square to make the cascades happen [...] This is sort-of proof that this is a minimalist addition to the game: I was able to implement it without changing or adding an entire system, I just needed to extend one that already existed.”

(Journal 2024-11-04.15)

Ultimately, this feature enabled the exploration of several important heuristics. Consequence, through new reactions to a single click and questions on how to communicate that clearly. Predictability and uncertainty, balancing the need for clarity in communicating the predicted effect of a click and the complexity of the game’s reactions to that click. Variety, as cascading chains created more options in level configurations. Narrow decision spaces, since the player’s input remained simple despite the system’s more complex output. Finally, abstraction, in how this complex cascading behavior should be communicated through minimal visual clues.

Cascading Squares into the Progression

After integrating the cascading squares feature, I focused on integrating it gradually into the game’s progression system. The simple randomized possibility of squares that could cascade created an unexpectedly steep difficulty spike.

“I had a player test the game for the first time today and the progression was very satisfying up until the cascading squares were introduced! That point is a way too steep difficulty spike, especially because the cascading squares can be of any targeting scheme, making it way harder to understand the feature if it's introduced on a complex targeting scheme rather than a simple one.”

(Journal 2024-12-05.3)

“Cascading toggles can trigger others in succession. This is good. This is a piece of emergent design that I didn't anticipate and it opens up complex player anticipatory play without me planning for it. This should be accounted for in the progression and ‘short’ cascades should be generated/introduced before ‘long’ ones.”

(Journal 2024-11-04.14)

This realization led to the decision to make cascading a progressive mechanic, introduced slowly across levels. I began by limiting the number of cascading squares allowed per level, gradually increasing the number as players advanced. I also made it so early levels wouldn't allow adjacent cascading squares, while later ones could feature connected, more complex, cascades.

Another design change came from observing misleading signifiers. Some squares were marked as cascading even though they could never actually be triggered by another square. This created false affordances, a visual promise of consequence that could never be fulfilled.

“Removed the possibility for squares that aren't targeted by others to be cascading [...] I wanted to exclude superfluous features like squares indicated as cascading that can't cascade because they can't be targeted by other squares. This is important because having a cascading indicator on something that can't cascade feels like a false signifier [...] the promise of well communicated, discoverable, and usable interactive design would be somewhat broken.”

(Commits 2024-11-15.1-3)

I also adjusted the level generation algorithm to control the introduction of cascading mechanics alongside other progression parameters, such as array size and targeting scheme complexity. With this adjustment, when cascades first appeared, the game temporarily returned to shorter

arrays and simpler arrow targeting schemes, reintroducing new complex features only once the player had completed a higher number of levels.

“Tweaked progression to introduce cascading toggles on less complex targeting schemes first [...] My intentions here were again to smooth the difficulty curve of the game so that when the hardest feature of the game is introduced, it's not coupled with some other hard features.”

(Commits 2024-12-09.1-2)

From a heuristic standpoint, the work to integrate the cascading squares into the progression brought to light the tension between predictability and uncertainty, trying to keep the system's complexity manageable by players while preserving surprise, discovery and ongoing challenge. Procedural generation was also a central concern, as integrating the feature into the difficulty curve and the game's content creation mechanisms revolved around managing the algorithm's constraints and adding new ones. Also, by constraining the number of simultaneous cascades and their connection, I also worked to maintain narrow decision spaces, making sure the players had space to think clearly within increasingly complex pairs of actions and reactions.

Single-Solution as the Core Game Mode

After lots of testing and iterations on both single-solution and multi-solution modes, I decided to make the single-solution mode the main mode of the game.

Players commented that single-solution mode felt clearer and more satisfying, offering stronger structure and sharper feedback loops. They also appreciated the way it established clear boundaries for understanding the system's logic:

“Something inherently more satisfying in the red-shaking response in single-solution mode. More like there are some hard edges players can use to try to understand the ‘world.’”

(Feedback 2024-11-06.3)

“Single-solution mode has great feedback when busting the clicks counter.”

(Feedback 2024-11-08.12)

This showed that this mode provided a sense of consequence, while multi-solution mode, though conceptually interesting, introduced ambiguity that weakened the connection between actions and reactions and even closure after solving a puzzle.

I considered keeping both, but my choice was guided by the game's minimalist philosophy: fewer parts, clearer goals, and tighter decision spaces. While I kept multi-solution mode in the back of my mind as a potential variant and maybe even a progression feature, I ultimately decided to prioritize consistency and wholeness, making every level adhere to a single clear objective.

No All-Self-Targeting Levels

This small but meaningful change was to prevent the level generation algorithm to output levels, outside the first one, to be built entirely out of self-targeting squares.

“I also tweaked a few things [so] that there can't be level[s] with all self-targeting squares (except the first level) because even though the first levels are meant to be simple, that's just plain boring.”

(Commits 2024-12-05.5)

This change ensured that even early puzzles maintained a degree of uncertainty, avoiding trivial arrangements. It strengthened the procedural generation system's capacity to keep the player engaged with minimal constraints.

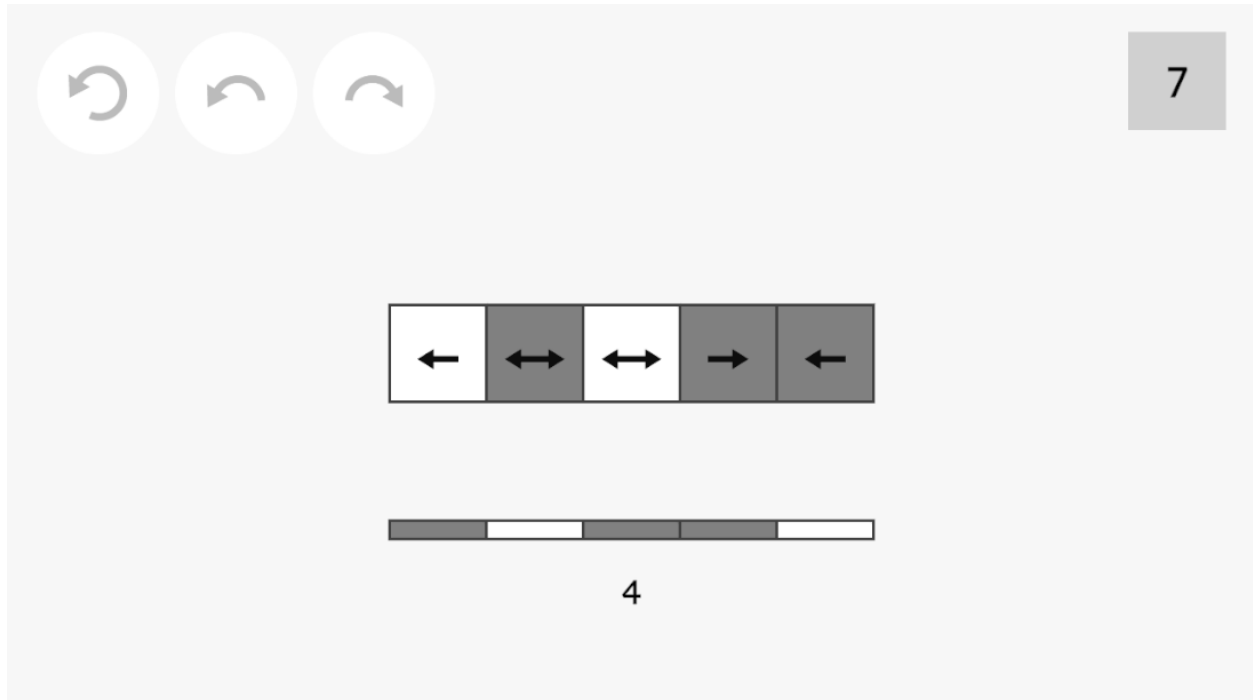
Removed the Trophy Icon

At one point during the development of *Toggle*, I added a trophy icon beside the completed levels counter to help players understand its meaning. While this recognizable symbol did help in clarifying the counter's function, I soon removed it, as it conflicted with the game's minimalist goal of non-representation and avoiding direct references to real-world objects.

“At some point previously, I attempted to create a connection between beating a level and the levels cleared counter by putting a known symbol (a trophy) next to the counter.

It proved useful but I quickly removed it when I realized it broke the minimalist intent of non-representation.”

(Journal 2024-12-12.5)



(Figure 19, *Toggle*, commit #23f72f9)

This change exposed the tension between predictability and abstraction. Removing the icon preserved the minimalist aesthetic but doing so made the interface lose some informativeness. The challenge remained to communicate meaning and relationships between this element of the game’s interface and player actions without relying on external symbols. I would have to find another way.

Prediction Dots

I replaced the color-coded prediction rectangles that showed which squares would toggle and in which state with minimal black dots. This new cryptic visual signifier conveyed less information but maintained a level of feedforward. This change originated from reflections about the balance of **predictability** and **uncertainty**:

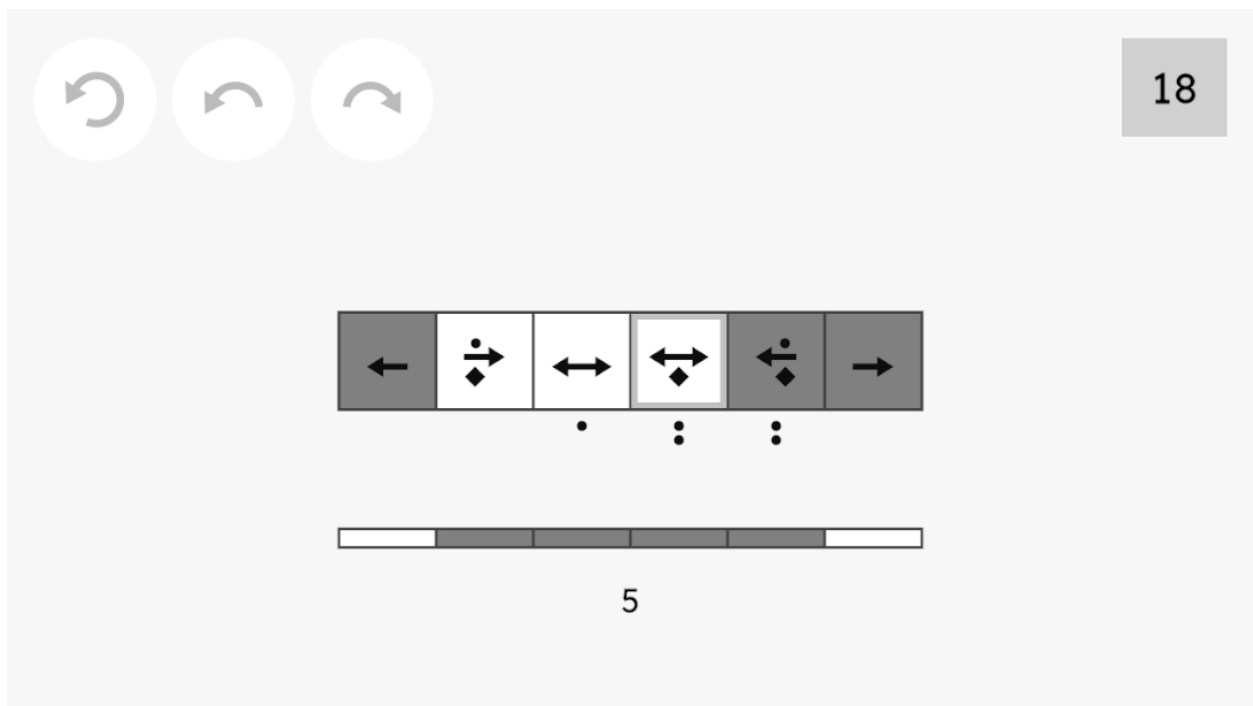
“I was giving away too much information with the prediction rectangles that helped predict in which toggled state would be a square if you pressed the currently hovered one.”

(Journal 2024-12-09.6)

The new dots were toggle-color agnostic, showing only which squares will be affected, not how. This reintroduced a layer of uncertainty, encouraging players to reason through the system rather than only follow visual clues.

“By changing it to a very austere black dot instead of a colored rectangle, I made it so players still had a hint at knowing which square would be toggled by the other squares, but they still had to ‘do the work’ of understanding the end color of the toggled square.”

(Journal 2024-12-09.7)



(Figure 20, *Toggle*, commit #478c8ff)

The small size and extreme abstraction of the dots also enabled an elegant, emergent benefit. Because they could now stack, they supported another kind of predictability: the visualization of chains of cascading toggles.

With the prediction dots, I found that a single flexible symbol could communicate complex information playfully. These dots were a great example of the principle of abstraction, while they also provided some predictability and supported the playful heuristic of uncertainty.

Visual Punch Animations

Originally, the game's interface was entirely static, presenting no motion, only instantaneous state changes. This aligned with the minimalist intent of avoiding unnecessary flourish, but as noted in my development journal, it turned out to be “satisfying for me, as a minimalist designer, but not very satisfying for the players.” Many players reported not noticing important visual elements or failing to connect them to the correct game mechanisms.

To address this, I introduced small “punch” animations on the clicks counter and levels cleared counter, two UI elements that were essential for understanding the game. As noted in a development commit:

“I [...] added a little punch animation to make sure the player notices it when it changes, so its function is more obvious. This should help players learn the mechanisms of the game quicker.”

(Commits 2024-12-11.3-4)

By timing these subtle movements with player actions (clicking a square or clearing a level) the animations acted as causal mappings, linking the player input on one element of the interface to another, informative element on the UI. The intent was to do so without introducing symbolic or metaphorical imagery. This approach preserves the minimalist abstraction while making the relationship between action and consequence more perceptible.

Hiding UI Elements

To reduce initial confusion, I implemented a system that hides certain UI elements until the player's actions make them relevant. For example, the undo, redo, and reset buttons remain invisible until the first move is made, and the levels cleared counter is hidden until the first level has been completed. As described in my notes:

“When a new player plays for the first time, there is a good chance they'll be very confused about some of the elements they see on screen for the first time. [...] So I continued making presentational changes to reinforce action/reaction mappings for these elements: hiding things (UI elements) until the specific point in time when it's used.”

(Journal 2025-01-03.1-2-3)

The intention here was to isolate the core interaction of the game (clicking squares) by reducing visual distractions and guiding players toward the only meaningful action they could take. This was a response to playtests where players hesitated or misinterpreted UI components:

“Some time ago I saw some players hesitate and try and click all of the buttons that appeared on screen, including the UI buttons. To minimize that, I'm trying to hide them before the first square is clicked. I hope that this communicates to the player that those buttons are secondary to the middle buttons and that they are not integral to the gameplay.”

(Commits 2024-12-27.7)

Additionally, revealing the levels cleared counter only after the first victory, and animating it from 0 to 1, helped communicate its function more clearly through cause and effect.



1

(Figure 21, *Toggle*, commit #28d3215)

This change reflects a balance between choice and predictability. By limiting the number of available actions in the beginning, I was able to establish clearer action-reaction mappings and guide players through narrow decision spaces. It also shows how abstraction can be used to make presentational decisions. Instead of relying on complex visual elements or representational clues, the game can communicate consequence through the revealing and animating of information at meaningful moments.

Changing the Position of the Goal Array

To help players more intuitively understand the objective of each level, I adjusted the layout by bringing the goal array closer to the interactive square array. This change was in response to recurring confusion about what the lower array represented. As noted in my journal:

“The objective array has frequently been associated with wildly different things, other things than just ‘showing the objective of the level.’ So I continued making presentational changes to reinforce action/reaction mappings for these elements: squeezing the objective array closer to the center interactive array to try and solidify the relation/mapping between a middle square and its equivalent objective rectangle.”

(Journal 2025-01-03.1-2-4)

Initially, I experimented with more explicit visual connections such as lines linking the arrays or sticking the goal elements directly beneath each square. However, these solutions felt too heavy-handed or broke the minimalist visual balance. Simply reducing the distance between the two arrays was a compromise I wanted to try out.

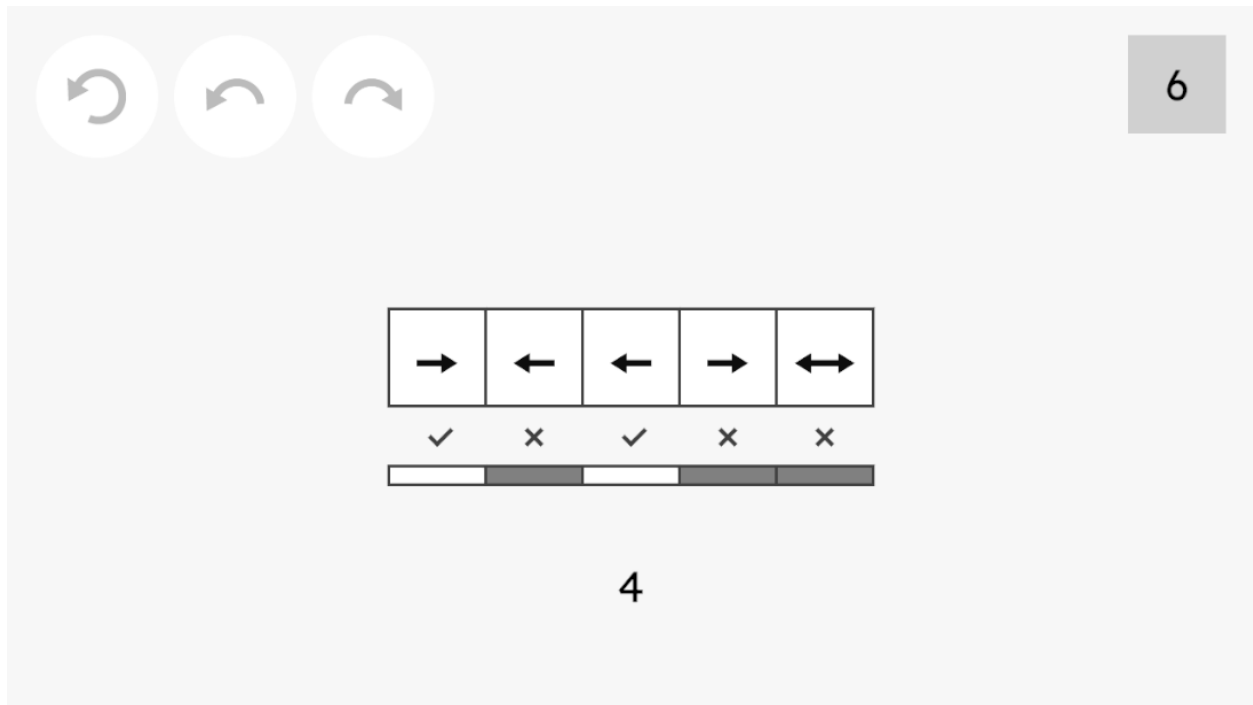
This small adjustment was made in order to improve mappings between different elements of the interface to the benefit of predictability. It also highlighted the heuristic of satisfaction by clarifying the player's sense of understanding and being able to reach the goal of the game. This was supported by wholeness, bringing separate parts together into a whole, and by abstraction, attempting to solve a communication issue through layout rather than representational clues

Crosses and Checkmarks

To continue making the goal of each level clearer and help players understand the relationship between their actions and the objective, I added crosses and checkmarks indicators between each interactive square and its corresponding goal rectangle. Updating with each new game state, a cross indicated that the square was not in the correct state to complete the level, while a checkmark confirmed that it was. As I wrote in my journal:

“Adding checks and crosses to hammer down on the mapping between the same elements mentioned above. These ‘evaluation graphics’ will also help in understanding their relation to the objective of the game. In adding these symbols, I’m getting dangerously close to breaking the non-representation minimalistic constraint. But I figured I can get away with using abstract symbolism for binary information (good or bad) as opposed to using images of real-world objects to represent complex ideas.”

(Journal 2025-01-03.5)



(Figure 22, *Toggle*, commit #c47288a)

The symbols served as immediate visual feedback, reinforcing the cause-and-effect relationship between player actions and progress toward the goal, while maintaining a level of abstraction that aligned with the minimalist visual language of the game. Even with these changes, however, later feedback suggested that understanding the mappings was still not perfect:

“Even with the new animations, checks and crosses to emphasize mappings between different elements, it’s still not great.”

(Commits 2025-01-27.2)

This highlighted the still ongoing tension between predictability, consequence, satisfaction and abstraction. Achieving balance between making the rules of the game (including its win condition) and player actions readable while preserving minimalism in the interface remained a delicate part of the design process.

More Visual Punch Animations

To strengthen the connections between player actions and their consequences even more, I integrated visual punch animations that happened whenever a square is toggled. These

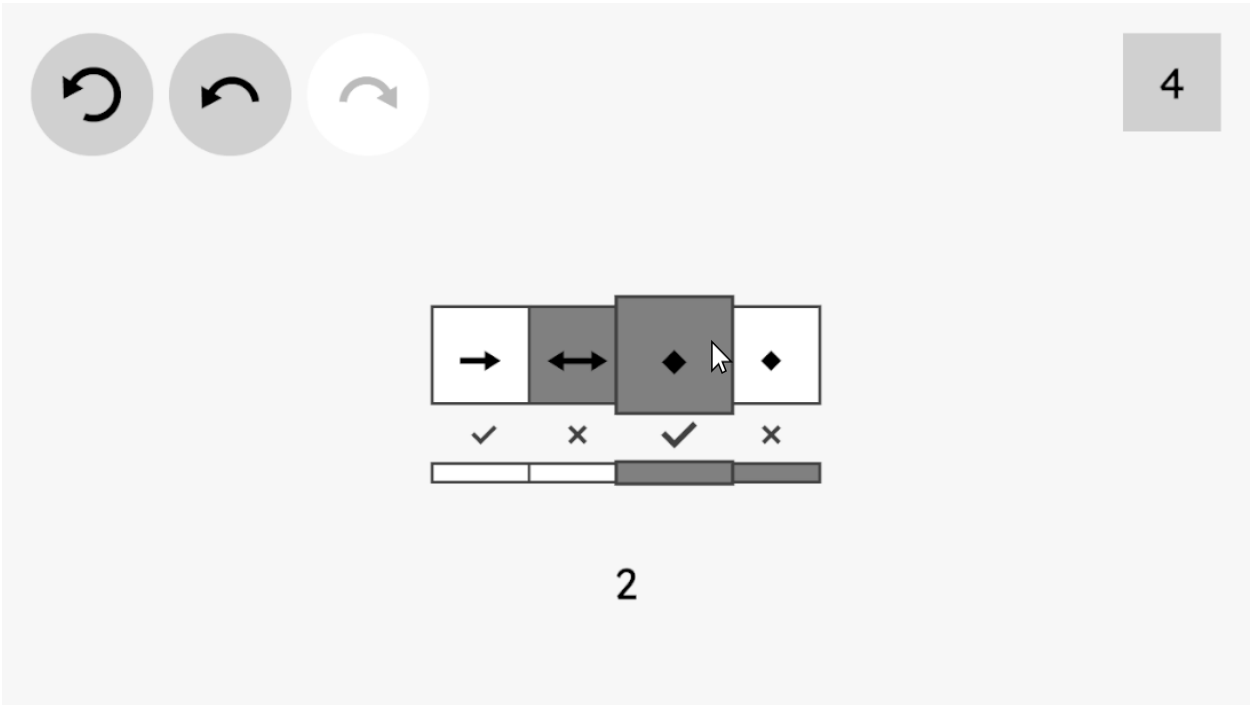
animations now happen simultaneously on three linked elements: each of the toggled squares, their corresponding goal rectangles, and the checks or crosses positioned between them. As I noted in my development journal:

“Squares now ‘bump’ when toggled, along with their respective solution rectangle and even with the associated check or cross. With this, every connected object of the game animates at the same time, connecting the action the player has done to the game’s reaction in a much more visually informative way.”

(Commits 2025-01-03.5)

“The more I integrate those animations, the more I’m realizing the power of ‘timing’ animations and punches to relate elements to each other and to player action (combined with the power of interactive causality).”

(Journal 2025-01-03.6)



(Figure 23, *Toggle*, commit #1f3e075)

By synchronizing the animated emphasis of these related elements, this creates a visually unified reaction to player action that clarifies how parts of the interface are connected

functionally. The synchronicity of the animations becomes a minimalistic communication tool, connected motion that expresses cause and effect through timing.

This iteration reinforces the heuristics of consequence and wholeness, using abstraction as a way to communicate relationships through motion rather than representation, turning this minimalist constraint into an expressive, dynamic form of feedback.

Nudge Animations and Delays

I wanted to clarify relationships within the interface even more and make cascades more readable. In order to do so, I introduced visual nudge animations and timed animation delays to improve the player's perception of how their click propagates through connected elements.

Soon after the implementation of the cascading toggles, I realized that the absence of movement made the mechanic difficult to parse:

“How do we show [the] resolution of multiple simultaneous toggles? It has to be a chain of animated feedback, not just all happening at once. Otherwise, there's no way the player is going to understand the 'chain' of toggles.”

(Journal 2024-11-04.3)

Later, reactions from players confirmed that instantaneous animations made cause and effect very hard to understand:

“The animation all happening at once is making it really hard to read what's going on. Being so rigidly minimalistic about not including movement might hinder the experience of learning the game for new players.”

(Feedback 2025-01-24.1)

This led to the introduction of staggered timing and small motion as clues. I made the arrows that indicate which squares will toggle briefly nudge forward, as if “pushing” toward their target squares. Each subsequent toggle in a cascade was also delayed a little but, producing a more legible sequence of cause and effect. I even delayed the level completion animation to capture the attention of the player sequentially, guiding the player's focus from one reaction of the game

to the next. Basically, I used these motions and delays as minimalistic expressive tools, making the game's visual language clearer without breaking the abstraction constraint.

“Even a small delay between the multiple effects of a click makes it so players can focus on each effect separately and can therefore better understand the cause and effect relationship between their actions and the reaction of the game elements.”

(Journal 2025-02-05.2)

“With that animation taking center stage after clicking a square, the ‘targeting of other squares’ effect is much clearer.”

(Journal 2025-02-05.4)

This addition shows how the heuristics of consequence, and abstraction can align again by making relationships more perceptible through timing rather than explicit description or symbolism. While movement does create tension with static quality the heuristic of wholeness suggests, it became a very useful minimal tool for clarity and expressiveness.

Removal of Prediction Dots and Crosses and Checkmarks

This last major change marked a significant turn toward refining the interface of the game through subtraction. The prediction dots and the checks and crosses were both originally conceived as visual clues to help players better perceive cause and consequence between their actions and the affected game elements. Over time, however, playtesting revealed that these additions did the opposite: they added noise, redundancy, and confusion.

“The game might be too maximalistic, the prediction dots add visual noise and might actually be distracting from the actual rules/mechanics of the game.”

(Feedback 2025-01-24.2)

After more iterations or the addition of other feedback systems, I eventually realized that the growing density of information the game was showing was counterproductive.

Removing the prediction dots was the first step. With the nudge animations and timing delays that made cause-and-effect more readable, the predictive graphics were no longer an absolute

necessity. On the contrary, they made players question whether or not there was another hidden feature related to these dots and created more confusion. These elements, originally introduced to aid the player, were actually introducing false affordances that players tried to integrate into their mental model of the game

“Someone told me during last playtests that, even though they were meant to help in understanding the mechanics and relationship between the squares, they added noise to the screen and made things muddier. [...] By adding the nudge animations and delays, these dots wouldn't be necessary anyway.”

(Commits 2025-01-27.5)

“They ended up complexifying the game by adding extra unnecessary information for the player to try and fit into their mental model. [...] Even though these dots were meant to simplify, they made the game not minimalistic enough.”

(Journal 2025-02-05.5)

Shortly after, I also removed the crosses and checkmarks, as they were found to overexplain, dilute the aesthetic clarity of the interface, and again create more confusion for the players that did not immediately grasp their relationship with their connected elements.

“I had made the decision to introduce [...] checks and crosses to try and help players understand the mechanisms making up the game. But it turns out that they [...] hindered their building of a mental model for the game”

(Journal 2025-02-05.7)

“Checks and crosses feel like too much handholding, it's also redundant information.”

(Feedback 2025-01-31.1)

The decision to take these elements away was not only a way to simplify the interface of the game, but a reaffirmation of what I wanted the game to be: a system of visible, discoverable cause-and-effect relationships that reward observation and interaction through experimentation.

“By clearing out the clutter, it also had the consequence of me leaning 100% into the minimalist intention of giving away as little information as possible to the player. For their enjoyment, of course.”

(Journal 2025-02-05.7)

By removing these visual elements, it exposed the connection between the heuristics of abstraction, wholeness, predictability, and consequence. At first glance, these principles seem to pull in opposite directions: the more abstract and reduced the interface becomes, the harder it could be for players to understand what's going on. But by removing redundant information, the game's direct feedforward and feedback became clearer. This kind of abstraction allowed the system to feel unified, whole and more readable as a coherent mechanism.

This process also revealed the relationship between narrow decision spaces, predictability, and consequence. When too much information was present on screen, players felt overwhelmed by decisions that didn't actually need to be made. By narrowing the visible and conceptual space offered to the players, I eliminated unnecessary distractions, allowing players to focus on the core, minimal set of interactions that truly mattered.

Finally, this change made me reflect on the connection between abstraction and embracing the heuristic of uncertainty. At some point during playtests, it became clear that players reacted very differently to the game's lack of explicit instruction or feedback.

“There's a clear distinction between two types of players: the ones that don't feel very engaged with the game because they don't understand what's going on and there's not enough feedback and information to get them going; and the ones that feel very stimulated by the lack of information and the motivation they feel to figure out what's going on and what to do with minimal feedback.”

(Feedback 2025-01-31.3)

For some players, the uncertainty created by a lack of information became a source of motivation. By leaning into abstraction to the point of removing all these indicators, I not only decluttered the screen but also fully embraced uncertainty to maximize the engagement of this type of player.

Goal Array as Squares

The last change I made to the game was a return to square shapes for the goal array. For most of the project, the goal indicators had been rectangles, intended to fit under each interactive square, but this created perceptual confusion. Players often interpreted the thin rectangles as something else than the objective of the level.

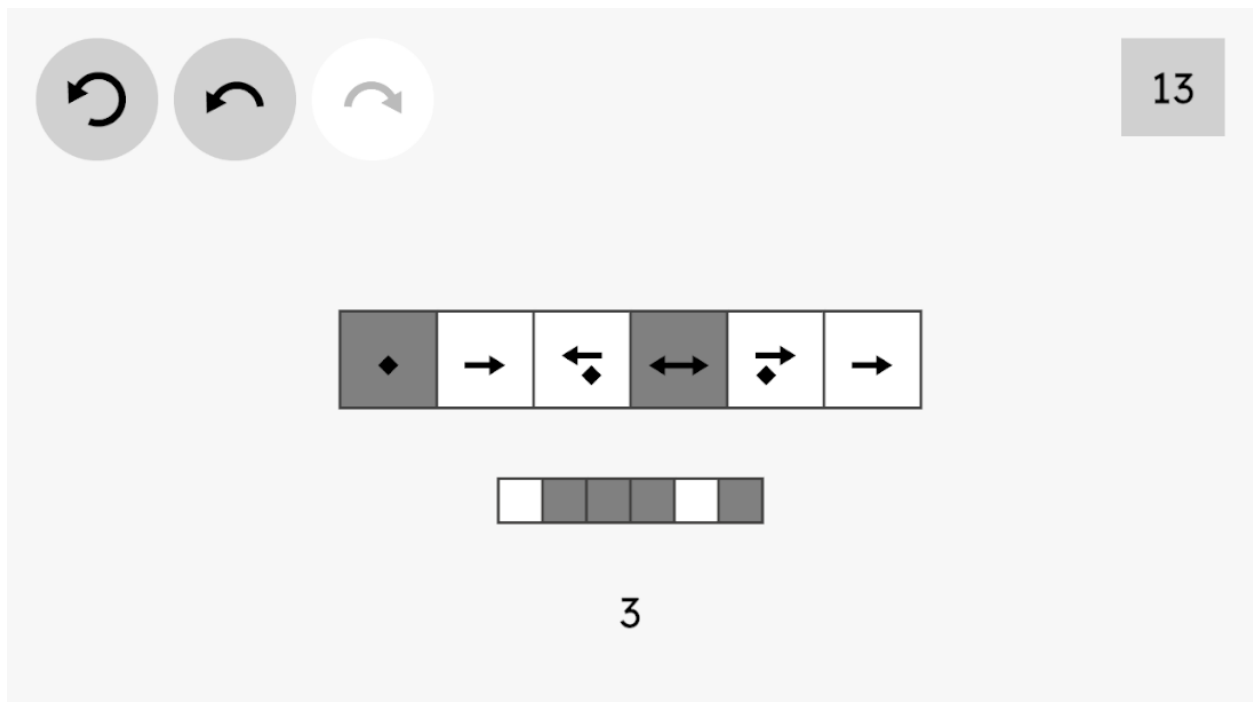
“Square shaped objective array feels better than the thin rectangles for ‘mapping’/associating it with the interactive middle array.”

(Feedback 2025-01-31.2)

Reverting to squares improved the perceived connection between the player’s interactive squares and the goal array. It emphasized shape-based mapping rather than relying solely on alignment, helping to connect the arrays in a coherent relationship.

“Both arrays being the same shape makes for a better mapping between the two elements. They basically feel ‘more like the same array.’”

(Journal 2025-02-05.9)



(Figure 24, *Toggle*, commit #41ff233)

This seemingly small change supported the heuristics of predictability (the player could more easily understand which goal square corresponded to which interactive square), satisfaction (the visual consistency reinforced the player's understanding of the goal of the game), and wholeness (the arrays were now part of a visually unified whole, as opposed to two separate and different parts of the system).

Points of Convergence and Tension

Through the development and design process of *Toggle*, the two sets of heuristics, those for playfulness and those for minimalism, interacted with each other. Sometimes they showed that concerns that emerged throughout development did not align. Some other times, they showed that some constraints reinforced each other. As we saw in this analysis, exploring these points of convergence and friction became central to the design process.

Here are the points of convergence and of tension that were examined through the two sets of heuristics and the reflection on the design process of *Toggle* that they brought to light:

Convergence

Narrow decision spaces ↔ Predictability Choice

- Fewer, clearer options to choose from reduce confusion while preserving the perception of agency through choice and play.

Narrow decision spaces ↔ Consequence Satisfaction

- Fewer options enhances the significance of each action, creating a strong sense of consequence and eventually closure, making even small variations in outcome feel meaningful.

Narrow decision spaces ↔ Consequence Wholeness

- Reducing the amount of decisions to be made by players maintains a closed, legible gameplay loop of action and feedback. This also reinforces the feeling of a coherent, self-contained system.

Procedural generation ↔ Variety

- Expanding algorithmic parameters increases the diversity of possible level configurations. Replayability and systemic diversity are supported when parameters are carefully tuned.

Procedural generation ↔ Wholeness

- Carefully structuring generative rules preserves a sense of coherence, reducing the likelihood of exceptions or one-off results that break the system's continuity.

Abstraction ↔ Consequence

- Even minimal or extremely abstract feedback can carry strong causal meaning. Synchronized motion, animation delays, and carefully choosing when certain visual elements appear can express direct relationships between different elements of the interface and between cause and effect. Abstraction also removes clutter so causality can be exposed more directly.

Abstraction ↔ Narrow decision spaces Wholeness

- Removing redundant or cluttered elements from the interface clarifies the mechanical structure for the player. This helps them merge multiple components into a single, whole perception of the system.

Abstraction ↔ Narrow decision spaces Predictability Consequence

- Simplifying the interface allows players to focus their attention on essential causal relationships. A system with fewer states and simpler abstractions make outcomes more predictable and their effects on future states more understandable.

Abstraction ↔ Uncertainty

- Minimal feedforward and feedback create an engaging space for curiosity and experimentation for players that feel satisfaction from deciphering the system themselves.

Abstraction ↔ Satisfaction

- Minimalist visual effects like small color shifts, timing differences, or pulsing motions can generate genuine feelings of progress or completion without explicit representational indications of success.

Abstraction ⇔ Predictability

- The more abstract and simplified a signifier is, the less it is bound to specific game objects or functions. This detachment makes it more adaptable, allowing it to convey a wider range of information, enabling more flexibility in predictability.

Wholeness ⇔ Predictability Satisfaction

- Reducing visual and structural differences between elements of the interface that share a connection enhances general coherence. This in turn helps players perceive relations between the game's parts and their connection with the goal.

Variety ⇔ Predictability Consequence Satisfaction

- Controlled variation helps create a rhythm for learning and gives meaning to progress, using diversity and discovery of content as a form of feedback and reward.

Tension

Narrow decision spaces ⇔ Procedural generation Uncertainty Variety

- Overconstraining algorithms undermines surprise, experimentation and exploration. We can loosen control over those constraints and restore variety, but it can lead to a loss in the generation of interesting results.

Procedural generation ⇔ Uncertainty Consequence Choice

- Randomization without control can generate trivial and unsatisfying level configurations where outcomes feel arbitrary or where player choice is irrelevant.

Abstraction ⇔ Predictability Consequence

- The more symbolic or reduced the visual language, the harder it becomes to ensure that players understand what elements mean or how they relate causally to each other. The lack of real-world clues makes learning more difficult, as players must deduce the rules without familiar references.

Abstraction ⇔ Satisfaction

- Ambiguity in minimalistic feedback can weaken the sense of closure when reaching a goal. If we reject too many rewards or goal structures in the pursuit of minimalism, players may feel limited in their satisfaction.

Variety ⇔ Wholeness

- Increasing the variety of possible interactions also increases the risk of undermining visual and structural consistency. This creates a need for trade-offs between content diversity and cohesion.

Consequence ⇔ Wholeness

- When player actions trigger multiple, indirect effects, the feeling of playing within a singular, unified system can be lost.

Predictability ⇔ Uncertainty

- More information makes interactions more legible but removes the mystery and imaginative space that can fuel play and discovery. These two heuristics exist in constant tension. While uncertainty provides meaningful opportunities for experimentation and learning, excessive uncertainty leads to confusion and frustration.

Synthesis

Across all features and iterations, the development of *Toggle* became an exploration of how meaningful, discoverable and legible play can emerge from a system built on minimal elements.

The minimalist game design process examined here has revealed a dynamic balance between legibility and mystery, constraint and variety, unity and affordance. In some iterations, adjustments, and feature implementations, these concerns converged. Viewed through the dual lens of heuristics for minimalism and playfulness, these moments showed how clarity and coherence can emerge from a small set of playful actions, each linked to simple but rewarding outcomes. Some other times, applying the two complementary sets of heuristic I proposed here

highlighted points of tension where too little, or too much, undermined the possibility of playful discovery or satisfaction.

What *Toggle*'s development demonstrates is that the work of minimalism does not reside in reduction alone, but in the constant rebalancing of how much is available for play, between what is shown and what is left for the player to infer, between the game's logic and the player's imagination.

Future Directions

In this project, the heuristics used to analyze the design process emerged only after development was complete. They crystallized through reflection and rereading my notes, revisiting design decisions, and tracing connections between tacit creative concerns and ideas drawn from game design, interaction design, and Minimalism as an artform. As a result, the heuristics served primarily as a retrospective analytical lens, helping reveal what shaped the game's evolution even when those influences were not consciously articulated during creation.

If I were to undertake a similar investigation in the future, I would take a different approach: establishing the heuristics before beginning the design process and treating them not only as interpretive tools but also as active constraints. Rather than uncovering themes after the fact, the heuristics could function as guiding principles and as a framework through which design decisions are filtered, questioned, and reframed. This would allow them to operate proactively, supporting reflection at the moment decisions are made rather than in retrospect.

This shift would open the possibility for a comparative study between two modes of research:

- Retroactive analysis, where using heuristics as analytical lenses reveals what mattered during creation, exposing patterns, tensions, and tacit priorities that emerged naturally through practice.
- Proactive constraint-driven design, where heuristics shape the creative process from the outset, emphasizing intentionality and structuring how constraints are applied, negotiated, or resisted throughout development.

Comparing these approaches would offer deeper insight into the role of these heuristics in minimalist game design and into how these reflective frameworks can support creative practice. This kind of future work would not only expand the methodological contribution of this project but also refine our understanding of how minimalist constraints can be actively leveraged to cultivate meaningful, legible, and playful experiences within intentionally limited systems.

Bibliography

Batchelor, D. (1997). *Minimalism* (S. Wilson, Ed.). Tate Pub Ltd.

Judd, D. (1965). *Specific Objects*. Arts Yearbook, 8.

Juul, J. (2007). *A Certain Level of Abstraction*. Situated Play: DiGRA 2007 Conference Proceedings, 510–515. <http://www.jesperjuul.net/text/acertainlevel>

Khaled, Rilla, & Pippin Barr. 2023. Generative Logics and Conceptual Clicks: A Case Study of the Method for Design Materialization. *Design Issues* 39 (1): 55–69. https://doi.org/10.1162/desi_a_00706

Khaled, R., Lessard, J., & Barr, P. (2018). Documenting trajectories in design space: A methodology for applied game design research. In *Proceedings of the 13th International Conference on the Foundations of Digital Games*, 1–10. FDG '18. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3235765.3235767>

Koster, R. (2004). *A Theory of Fun for Game Design*. O'Reilly Media, Inc.

Meyer, J. (2004). *Minimalism: Art and Polemics in the Sixties* (Illustré édition). Yale University Press.

Morris, R. (1966). *Notes on Sculpture*. Artforum. <https://www.artforum.com/features/notes-on-sculpture-211706/>

Myers, D. (2009). *In search of a minimalist game*. 2009: Proceedings of DiGRA 2009 Conference: Breaking New Ground: Innovation in Games, Play, Practice and Theory. DiGRA 2009.

Nealen, A., Saltsman, A., & Boxerman, E. (2011). *Towards minimalist game design*. Proceedings of the 6th International Conference on Foundations of Digital Games, 38–45. <https://doi.org/10.1145/2159365.2159371>

Norman, D. (2013). *The Design of Everyday Things: Revised and Expanded Edition* (Révisé édition). Basic Books.

Pajitnov, Alexey. (1988). *Tetris*. Mirrorsoft.

Perrault, J. (1967). *Minimal Art: Clearing the Air*. Village Voice.

Playdead. (2010). *Limbo*. Microsoft Game Studios.

Schön, D., & Bennett, J. (1997). Reflective Conversation with Materials. In *Bringing Design to Software*. 172–184. Addison-Wesley.

Strickland, E. (2000). *Minimalism: Origins*. Indiana University Press.

Superhot Team. (2016). *Superhot*. Superhot Team.

Svanæs, D. (2000). *Understanding Interactivity: Steps to a Phenomenology of Human-computer Interaction*. Norges teknisk-naturvitenskapelige universitet, Institutt for datateknikk og informasjonsvitenskap.

Upton, B. (2015). *The Aesthetic of Play* (1st edition). The MIT Press.

Wollheim, R. (1965). *Minimal Art*. Arts Magazine, January.

Zubek, R. (2020). *Elements of Game Design*. The MIT Press.