Reflective Creators

Max Kreminski and **Michael Mateas**
University of California, Santa Cruz
{mkremins, mmateas}@ucsc.edu

Abstract

Casual creators are a genre of *autotelic*, or process-focused, creativity support tools (CSTs) that prioritize the aesthetic experience of the creative process over that of the resulting product. Typically, casual creators aim to elicit a sense of *ease* and *pleasure* in their users. These are, however, not the only aesthetic experiences that a process-focused CST might reasonably aim to elicit. We introduce *process aesthetics* as an analytical lens through which to examine the motivations and values of different autotelic CSTs, including but not limited to casual creators. Based on this analysis, we then investigate a novel process aesthetic—*reflection*—for autotelic CSTs, present a set of autotelic CST design patterns intended to elicit reflection, and discuss three case studies of autotelic CSTs that make use of these design patterns.

Introduction

Casual creators are a genre of systems that "support creativity as an intrinsically pleasurable activity, rather than as an extrinsically-motivated way to accomplish tasks" (Compton and Mateas 2015). Many casual creators are based on computationally creative systems, resulting in a mixed-initiative co-creative user experience (Liapis et al. 2016). Because casual creators emphasize process over product, they are defined principally in terms of how they make their users feel during and after the creative process: exploring a creative possibility space with a casual creator should feel "fast, confident, and pleasurable", and users should experience "feelings of pride, ownership, and creativity" when they look back on the artifacts they have made (Compton and Mateas 2015). In essence, a creativity support tool (CST) can usefully be viewed as a casual creator if it primarily aims to elicit feelings of *ease* and *pleasure* in the user.

This experience-focused definition of casual creators raises the question of whether ease and pleasure are the only, or the most important, experiential qualities that a CST could aim to elicit in its user. We contend that they are not—and, more specifically, that a CST can support autotelic creative activity without necessarily being a casual creator. The subjective experience of creation can be worthwhile even when it is not centered on feelings of ease and pleasure, and a focus *exclusively* on ease and pleasure might obscure or interfere with the design of autotelic CSTs that target alternative experience goals.

One alternative experiential quality that an autotelic CST might aim to elicit is that of *reflection*. A reflective CST might be designed to support users in carefully considering the implications of their creative decisions; introspecting on what they choose to create, how they choose to create it, and why; or attempting to refine their creative goals, intuitions, or processes through the act of creating. Users of reflective CSTs might want to create as a form of meditation or as a way of working through their thoughts and feelings on a topic, with no intention of sharing (or even preserving) the things they create. They might want to explore a design space slowly and systematically, rather than rapidly and easily converging on a specific corner of the space. And they might want to bounce ideas off of an explicitly critical computational collaborator: a trusted adversary that they can count on to push back on their decisions, or ask them to justify their actions. For these users and use cases, casual creator design patterns might not be appropriate, even as the user's focus remains on the creative process rather than on the products that emerge.

In human-computer interaction, systems that attempt to make their users feel a specific way have been described both in terms of the "aesthetics" (Boehner, Sengers, and Warner 2008; Höök 2008) and "use qualities" (Löwgren and Stolterman 2004; Isbister and Höök 2009) they aim to elicit. In game design, the popular Mechanics-Dynamics-Aesthetics framework (Hunicke, LeBlanc, and Zubek 2004) uses the term "aesthetics" to refer to the experiential qualities of gameplay. In this paper, we will use the term *process aesthetics* to refer to the experiential qualities of CSTs: "aesthetics" in keeping with the terms used in adjacent fields for similar phenomena, and "process aesthetics" specifically to distinguish the aesthetics of *using* a CST from the aesthetics of the artifacts that a CST is used to create.

In the remainder of this paper, we first characterize reflection as a process aesthetic and argue that it is a viable aesthetic for autotelic CSTs. We then present a sampling of ten design patterns that existing CSTs use to elicit reflection. Finally, we discuss the application of these patterns to the design of three recent CSTs that embrace reflection as a design goal and conclude with suggestions for future work. We hope that our work will encourage the development of both deliberately reflective CSTs and additional new process aesthetics for CSTs beyond reflection.

Related Work

Reflection in Computational Creativity

To date, computational creativity research has primarily engaged with the concept of reflection by trying to construct computationally creative systems that are capable of reflecting on their own work. This approach to reflection dates back at least two decades, to Buchanan's (2001) proposal of a new goal for future work in computational creativity research: the construction of computational systems that can exhibit creativity by reflecting on their own programmed limitations and finding ways to surpass these limitations through self-modification.

In a similar vein, the computationally creative system MEXICA (Pérez y Pérez and Sharples 2001) is based on the engagement-reflection model of creative writing (Sharples 1999), which treats reflection as a key part of the creative process. As a generator of "story frameworks", MEXICA engages in a cyclic process of writing, reading what it has written, reflecting on what it has read to identify potential points of improvement, and rewriting with these points of improvement in mind.

The notion of reflection as an operation that takes place within a computational system, whereby the system reflects on its own work, also forms a key component of the Creative Systems Framework (Wiggins 2006). This framework has seen wide adoption within computational creativity research as a formalization of the field's goals.

In this paper, however, we are not primarily concerned with building computational systems that reflect on their own work. We are instead interested in building creativity support tools that can provoke reflection *in their human users*, regardless of what happens within the computer. For us, a "reflective creator"—a CST intended to elicit reflection—is successful if and only if its human users find themselves drawn into reflective contemplation of their own creative goals, practices, successes, and failures. This definition mirrors the established definition of the earlier term "casual creator": a CST that is judged successful if and only if its human users find it easy and pleasurable to use. In this sense, our view of reflection as a process aesthetic to be elicited in a human user—rather than an operation or routine that is carried out by a computational system—represents a departure from how reflection is typically viewed in computational creativity research.

Reflection as a Process Aesthetic

In *The Reflective Practitioner* (1983), Schön describes two forms of reflection with implications for creative practice: reflection-in-action (carried out in the moment, while a situation is still unfolding) and reflection-on-action (carried out retrospectively, once a situation has reached quiescence). Both forms of reflection are essential to the avoidance of creative impasses that result from an excessively narrow focus on one formulation of a creative problem. When a practitioner realizes that the scripts or techniques that they have been attempting to apply to a problematic situation are not yielding the desired results, it is through reflection that they can identify the essence of the mismatch, allowing them to reformulate their understanding of the problem (or of their own practices, tools, and professional role) in order to resolve the impasse. Per Compton and Mateas, casual creators are intended to expedite reflection-in-action that takes place on relatively short timescales—but reflection-on-action, and even reflection-in-action that plays out over a longer timescale, are both sidelined in the interest of keeping interactions fast and fluid. Our investigation of reflection as a process aesthetic was initially motivated by an interest in what an autotelic CST that supports reflection-on-action, or slower and more contemplative forms of reflection-in-action, might look like.

Smith (2017) surveys and critiques the design values that motivate the majority of research in procedural content generation today, while proposing reflection—as well as materiality and discomfort—as possible alternative values. In discussing the possible benefits of reflection, Smith echoes the call in HCI research for "slow technology" that deliberately prolongs interaction in order to imbue it with richer texture and deeper meaning (Hallnäs and Redström 2001).

Perhaps counterintuitively, support for reflection as an aesthetic—one that is not always compatible with ease and pleasure—can also be found in game design. Though early game design discourse often emphasized *fun* as the primary goal of design, recent years have seen an "opening up" of fun to reveal a broad spectrum of orthogonal or even contradictory possible aesthetics. Just as personal and queer games often deliberately reject fun in favor of alternative aesthetics, including discomfort and frustration (Anthropy 2012), autotelic CSTs could equally embrace process aesthetics beyond ease and pleasure. One could imagine, for example, a critical CST that highlights how design decisions impact human participants in the supply chain, just as Molleindustria's *The McDonald's Videogame*[1] seeks to emphasize the human impact of the business practices it models.

Autotelic Creativity Support Tools

Nakakoji (2006) divides creativity support tools into three categories: those analogous to *running shoes*, *dumbbells*, and *skis*. Running shoes aim to provide additional support for a well-understood activity (running) with obvious criteria for success; dumbbells are used to develop creative capacity in the user without being employed to produce creative artifacts directly; and skis attempt to enable a new form of creative activity (skiing) that would not be possible without the tool. Because skis are intended to enable new activities, they can be hard to evaluate initially if other tools enabling the same experience don't already exist. Process aesthetics present one possible strategy for evaluating tools of this nature in terms of whether they successfully enable an experience with particular subjective qualities.

Dumbbells hint at another possible justification for reflection as a process aesthetic. Krakauer (2016) extends Norman's description of computer systems as cognitive artifacts (Norman 1991) by drawing a distinction between *complementary* cognitive artifacts, which build up capabilities in the artifact's human users that remain even when the ar-

[1] http://www.molleindustria.org/mcdonalds/

tifact itself is removed, and *competitive* cognitive artifacts, which replace or displace the user's capabilities. From this perspective, reflective creators that can assist users in developing a tool-independent reflective creative practice could be viewed as complementary cognitive artifacts. However, this is not the only reason that reflective creators might be valuable. Like ease and pleasure, reflection can be an instrumental value for CST designers—judged as useful because it leads to the production of more or better artifacts or creators—but it can also be a terminal value, judged as inherently worthwhile.

In her dissertation, Compton (2019) discusses the difficulties associated with "slow creators", which have a wide gulf of evaluation due to inherent limitations in the speed of the underlying computationally creative system. However, this analysis does not consider the possible value of slowing down the creative process in order to promote reflection.

Petrovskaya, Deterding, and Colton (2020) survey existing commercially available casual creators and categorize them according to their main interaction technique. The findings of this survey seem to support the claim that most commercially available apps fitting the definition of casual creators are optimizing primarily for ease and pleasure, and especially for *speed* of creation in support of this goal.

Nelson et al. (2018) suggest that at least some users of casual creators are motivated primarily by curiosity, either about the tools themselves or about the generative spaces that these tools allow their users to access. The user behavior patterns documented in this work may imply that some users of casual creators actively seek out a reflective creative experience, rather than an easy or pleasurable one.

Design Patterns

How, concretely, can CSTs be designed to elicit reflection? In this paper, we identify an initial set of ten promising design patterns drawn from existing systems. Design patterns, as introduced by Alexander (1977), are high-level descriptions of solutions to problems that frequently recur in a particular design space; here, we present patterns that apply to the design space of CSTs and the recurring problem of eliciting reflection in the human users of these tools.

Like the accounting of casual creator design patterns provided by the original casual creators paper (Compton and Mateas 2015), this list of reflective creator design patterns is not intended to be exhaustive. Instead, we aim to represent features that are commonly found in existing reflection-focused CSTs, and that we have found useful in our own design analysis of CSTs (especially the three case studies discussed in the following section).

Reifying intent Many reflective creators ask their users to make their creative intent explicit and provide a set of mechanisms for describing and negotiating intent. Often—but not always—these tools provide an *intent language* (Martens and Hammer 2017) that allows users to describe their intent in a systematic, machine-parseable fashion.

Asking users to make their intent explicit can promote reflection even when the intent is not understood by the machine. In a multi-user context, reifying the design intent makes it a shared object of comment between the users and enables metaconversations about what an artifact should and should not include. And even in a single-user context, the mental work of identifying and expressing one's intent requires reflection on one's own goals, values, and priorities, regardless of whether the intent is then fed into a computational system.

One multi-user example of this pattern can be found in the tabletop storytelling game Microscope[2], which provides a feature called the *palette* that affords negotiation among players as to what they would and would not like to see happen in the story. Similarly, PolicyKit (Zhang, Hugh, and Bernstein 2020) provides internet communities with an intent language for describing and negotiating moderation policies, prompting members of these communities to reflect on and openly discuss how they want to be moderated and why. Here, however, the intent language is fully system-understandable, enabling the underlying "policy engine" to automatically enforce agreed-upon policies.

Elaborating intent Once the user has specified their intent in a systematic way, an obvious next step is to generate and display many possible realizations of that intent. In so doing, a computational tool can attempt to present the user with information on potentially unexpected ramifications of the intent; contradictions hidden in the intent; or discrepancies between the envisioned and actual consequences of the intent as directly specified.

This is central to the approach taken by the game design support tool Germinate (Kreminski et al. 2020c), which uses generative methods to translate a user's initial high-level rhetorical intent into a variety of playable digital games. A similar approach is taken by creative writing support tools Writing Buddy (Samuel, Mateas, and Wardrip-Fruin 2016) and *Why Are We Like This?* (Kreminski et al. 2020b); these tools both allow users to specify and modify their storytelling goals, then suggest possible character actions that might help to advance these goals.

Inferring intent Based on the creative decisions a user has made, the system can also attempt to *infer* their intent and display it to them—sometimes as a set of sentences in a formal intent language, sometimes merely as a list of adjective labels. The difficulty of translating an implicit creative intent (which the user does not yet fully understand) into a explicit intent can thereby be mitigated: it is usually easier to accept or reject specific system-suggested assertions about an intent than it is to write out an explicit statement of a formerly purely implicit intent from scratch.

Germinate infers intent from mixed user-specified and system-generated game rules via *proceduralist readings*, enabling it to suggest new high-level design goals to the user based on decisions they have already made.

Mahajan et al. (2019) offer automated critiques of student-created branching narrative projects by comparing the input project to a database of existing projects on a variety of descriptive criteria. Their system then produces a report for the student on which existing projects were most

[2] http://www.lamemage.com/microscope/

and least similar to theirs and on what criteria (including frequency of choices, density of text, and overall length) their project stands out relative to others. This report can be read as an inference of intent, allowing the student to understand how their project might come across to the reader and either lean into or modify their approach based on whether the inferred intent matches their preferred direction.

One promising technical approach to inferring intent involves the use of discriminative learning, as discussed by Karth and Smith (2019) and Kreminski, Wardrip-Fruin, and Mateas (2020). Based on a set of user-provided example artifacts, a co-creative system can infer intent from the shared characteristics of these examples, generate more examples based on the inferred intent, present the generated artifacts to the user, and let the user approve or reject the generated examples to progressively refine their intent.

Interpretive refraction One way to facilitate reflection is via defamiliarization, or deliberate creation of distance between creators and their practices or artifacts when they would otherwise be "too close" to see the flaws. To achieve this defamiliarization, it can be helpful to display to the user multiple different computational judgments or readings of the work-in-progress artifact, even when these readings seem to point in mutually incompatible directions.

In contrast to the *entertaining evaluations* often employed by casual creators, reflective creators more frequently employ evaluation methods that are intended to be taken seriously. Additionally, in order to ensure that real co-interpretation (Pousman et al. 2008) takes place between the user and the system, the system's readings should involve some actual analysis of the creative artifact or process; they should not be wholly disconnected from the user's input, as is sometimes the case in casual creators (such as *BECOME A GREAT ARTIST IN JUST 10 SECONDS*[3], which presents a humorous but baseless "similarity score" between the user's glitch-art creation and a famous classical painting.)

Smith et al. (2015) leverage interpretive refraction to defamiliarize the process of textile crafting. Their crafting tool sonifies the user's physical gestures to provide a new, unfamiliar feedback channel and thereby prompt reflection on practice. Also in the domain of physical fabrication, Fabricaide (Sethapakdi et al. 2021) prompts users to consider the material costs of their design by calculating how much of what materials would be needed to create it and visualizing this information in real time.

Sentient Sketchbook (Liapis, Yannakakis, and Togelius 2013) uses visualizations of different "fitness dimensions" to highlight different aspects of game design. This idea has been generalized somewhat in the context of game design support under the label of "computational critics" (Osborn, Grow, and Mateas 2013): systems that examine different aspects of a game design and offer criticisms.

Contextualizing choices In Schön's account of creative design, the design process involves "spinning out a web of [...] implications" (Schön 1983) in which each individual design decision may impose far-reaching constraints on other aspects of the design. Because these constraints have potentially nonlocal impact, and because designs are often so large that it is impossible to attend to all of a design's features and implications at once, it can be easy for design moves to quietly invalidate other choices that have been made during the design process, often in a way that is not immediately visible to the designer. Therefore, it may be especially important for reflection-focused CSTs to support their users by highlighting the implications of decisions.

For instance, if the computer is capable of understanding the impact of a single choice in terms of multiple different aspects of the high-level design intent, it can inform the user when a choice that appears to be a good one for local reasons (perhaps because it advances one particular design goal) also has negative impacts on the realization of other design goals, which the user might not currently be considering. If the design space is understood as containing choice points, as Schön suggests, then decisions that cut off certain parts of the design space might also be useful to highlight. This kind of feedback might be especially easy to provide when the underlying generative system already understands the creative process in terms of the navigation of a design space, as (for instance) in design space modeling approaches to procedural content generation (Smith and Mateas 2011). Finally, a system that understands the impact of individual moves might also use that information to show creative decisions in context, among other strong possibilities (for instance by prompting the user with alternative moves that they could reasonably have chosen to perform instead).

One example of this approach can be seen in the work of Kybartas, Verbrugge, and Lessard (2020): a co-creative narrative system that operationalizes the possible worlds theory of narrative structure to identify a "tension space" consisting of conflicts between characters' ideal world states and the state of the world as it actually exists. This tension space is then visualized as it evolves throughout the process of authoring an emergent narrative storyworld, making it apparent to the user when a creative decision that they have made substantially increases or reduces narrative tension.

Challenging choices One of the editor's roles in the creative writing process is to constructively push back against the writer's decisions. Similarly, some users deliberately seek out or implement CSTs that help them catch and eliminate their bad habits, for instance by automatically detecting and flagging uses of specific words or sentence structures.[4] Computationally creative systems with a deeper understanding of how creative artifacts are structured could extend this kind of reflection support to other domains.

Many co-creative systems already base their actions (or action suggestions to the user) on a model of what creative decision would be most likely at this point in the creative process. For instance, language model-based CSTs for writing often use likelihood to recommend words or phrases to the user (Manjavacas et al. 2017), while some game design tools do the same for level design decisions (Guzdial et al. 2019). This same information could be used to push

[3] https://igf.com/become-great-artist-just-10-seconds

[4] http://matt.might.net/articles/shell-scripts-for-passive-voice-weasel-words-duplicates/

users away from making clichéd decisions, for instance by informing them when they perform a highly likely action and suggesting less-likely alternatives.

Like the previous design pattern (**contextualizing choices**), this pattern can increase users' doubt in their own choices and slow their exploration of the design space considerably. As a result, both patterns might be considered undesirable in many casual creator contexts. In a reflective context, however, introducing doubt and slowdown may be exactly what is needed.

Reflective encoding Some CSTs that support the construction of generative models are sometimes used not to build realistic or directly useful models, but instead to assist users in reflecting on the phenomenon they are modeling. One example is the social simulation tool Ensemble (Samuel et al. 2015): on at least two occasions, it has been used to model real-world social phenomena with the goal of reflectively developing a better understanding of that phenomenon, but without regard for the direct applicability of the resulting model (Dickinson, Wardrip-Fruin, and Mateas 2017; DeKerlegand, Samuel, and Leichman 2020). Critically, even when these models are not immediately applicable to any existing problem, the act of constructing them—of formalizing knowledge sufficiently that it can be encoded in a relevant notation—prompts reflection and a deepening of understanding within the model's creator. This is reminiscent of the autotelic uses of formal languages (such as baseball scorecards) discussed by Nardi (1993): some baseball fans find that their understanding and enjoyment of the game is deepened when they follow along by reflectively encoding the action of the players into a formal language.

Though the generative text tool Tracery (Compton, Kybartas, and Mateas 2015) is often cited as an example of a casual creator, some of its more advanced features (such as actions, which allow generated substrings to be saved and reused) sometimes compromise the aesthetic of ease. However, these features are key to enabling the form of reflective encoding for which Tracery is often used: reflectively building up a Twitter bot or other text generator to imitate a particular corpus, essentially conducting a manual *generativist reading* of the corpus in question (Kreminski, Karth, and Wardrip-Fruin 2019).

Future encoding-focused reflective creators might prompt the user to incrementally flesh out the model they're building, perhaps by identifying underspecified parts of the model and asking pointed questions about them (as in Garbe's proposed worldbuilding assistant chatbot.[5])

Reflective enactment Some computationally creative systems, especially in the domain of textile crafts, generate designs that can only be physicalized by human labor. Though this could be seen as a weakness of these systems, Albaugh et al. (2020) suggest that the *underdetermination* of computer-generated designs can also be viewed as a resource for promoting creative reflection. Laborious enactment of computer-generated instructions can prompt reflection on the meaning and significance of the crafting process and the crafted artifact alike.

Embroidered Ephemera (Sullivan 2020) is a computationally creative system that generates an embroidery sampler design from a user-selected tweet, but leaves the work of actually embroidering a generated design to the user. Though the system was initially conceived of as a casual creator, its author reports that the time-consuming nature of embroidery work compromises the aesthetic of casual creation in some regards. We argue that this is because Embroidered Ephemera instead exemplifies an aesthetic of reflection in this aspect of its design. By contrasting the low time cost of selecting a tweet to feed into the system with the high time cost of embroidery, the system creates a moment of *reflective commitment* at the time of tweet selection, encouraging the user to carefully consider why they might want to physicalize this particular tweet in this particular way at this particular time.

Reflective repair Reflective enactment leverages the *incompleteness* of computer-generated designs to prompt reflection through the process of completing them. Reflective repair deepens this focus on incompleteness by introducing *incorrectness* to the computer-generated designs as well, requiring users of a computer-generated artifact to fix up minor problems or fill gaps left by the computer. Repair is an essential component of Sharples's model of reflection in creative writing (Sharples 1999), and encouraging users to engage in thoughtful repair of flawed artifacts may be an especially useful strategy for promoting the development of reflective capacity that persists in users beyond their experiences with a particular tool.

The SkyKnit system (Shane 2018), which uses machine learning to generate knitting instructions, provides a key example of reflective repair in action. Because the instructions that the system generates are often flawed, knitters who attempt to realize these designs are frequently forced to improvise repairs to nonsensical aspects of otherwise-acceptable plans. This can result in a productive kind of defamiliarization, challenging knitters to think outside the box and sometimes even invent new stitch types in their attempts to repair machine-generated designs.

Reflective revisitation Frequently during the design process, a designer's attention is fixed on one specific, narrow part of a larger, more complicated design situation. This can introduce problems when design decisions made in one part of the situation subtly invalidate decisions made elsewhere, but the designer has not yet noticed the conflict. To combat the tendency for earlier decisions to linger even when they no longer fit the design situation as a whole, a computational creative partner might explicitly prompt its user to re-engage with or reevaluate decisions they made in the past.

This might take several forms. Building on the pattern of **challenging choices**, the CST might actively seek out decisions that were made some time ago and prompt the user to reconsider the decision in light of how the design as a whole (including the design intent) has evolved since then. Additionally, a CST might gradually introduce and tighten artificially imposed constraints on the design in order to force periodic reevaluation of decisions; similar patterns are em-

[5] https://twitter.com/logodaedalus/status/919403844404051969

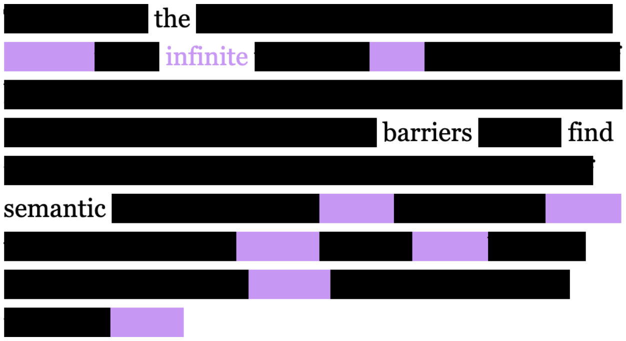


Figure 1: A screenshot of Redactionist, showing a block of input text mostly erased, with some user-selected words still visible and options for other user-selectable words highlighted in lavender. Hovering over a selectable word will reveal it, and clicking it will add it to the poem. Clicking an already-selected word will deselect it again, possibly opening up alternative words for selection instead.

ployed by puzzle games like SpaceChem[6] that encourage users to search for progressively more creative solutions to a fixed puzzle by imposing tighter and tighter budgets on the resources that can be employed in the puzzle's solution.

Effective deployment of reflective revisitation may sometimes require that the creative process be deliberately slowed down, in order to allow for meaningful distance between decisions to emerge. In particular, deliberately splitting the design process into multiple distinct sessions might help refresh the user from one session to the next, allowing them to revisit past decisions with a slightly different perspective.

Case Studies

To demonstrate the utility of a reflection-focused perspective on CST design, we apply our perspective to the analysis of three existing mixed-initiative CSTs created by the first author, all of which were initially framed as casual creators but did not seem to fit the category perfectly. In each of these cases, the reflective creators framework helps us deepen our understanding of how these systems work (from a user interaction perspective) and identify directions for further development.

Redactionist

Redactionist (Fig. 1) is a mixed-initiative CST for erasure poetry creation. A Redactionist poem is a short declarative sentence produced by erasing most of the words from a block of input text, retaining only words with appropriate part-of-speech tags and grammatical interrelationships to match one of a few dozen possible patterns—for instance, the pattern `ARTICLE NOUN VERB ARTICLE ADJECTIVE NOUN` would match sentences like "the poem conceals an elusive metaphor".

Redactionist went through three distinct stages of development. It initially took the form of a push-button generator called Blackout (Kreminski, Karth, and Wardrip-Fruin 2019) that, given some input text, would produce a poem by erasing words from the input without further user interaction. To make the system a better casual creator, it was then updated with a mixed-initiative interaction model. Given a block of input text, the system would scan the text from left to right and come up with three possible next words for the poem to include. Then it would present these options to the user, wait for a selection, and continue scanning left-to-right, repeatedly calculating three more options based on the user's previous choices until the poem was complete. Limiting the order of text traversal from left to right and restricting the user to three choices at each step was intended to limit overwhelm, in keeping with the prioritization of ease-of-use over fine-grained control in casual creator design.

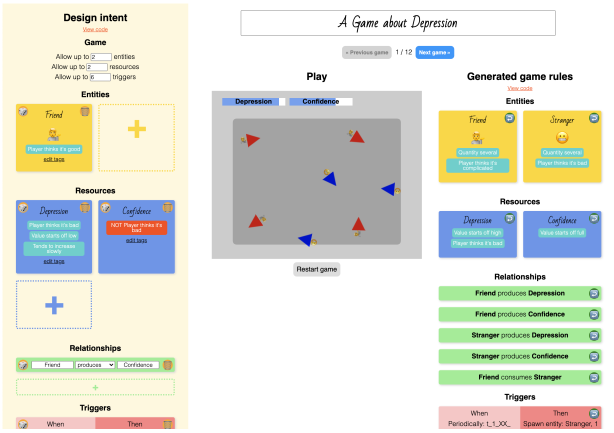
However, this version of the system was found to be unsatisfying. Despite the potential for greater overwhelm, removing some of the artificial restrictions on the possibility space of each poem turned out to produce better results, especially when combined with a deliberate application of the **contextualizing choices** design pattern. In the most recent version of Redactionist, the system identifies up front a set of words that are valid for inclusion in the poem and presents the user with the initial choice to select any one of these words. The selection of a word constrains what patterns might be viable matches for the selected set of words, restricting future choices somewhat. At any time, any selected word may be unselected again, possibly removing constraints and enabling some other words to be selected instead. This turned out to produce a compelling reflection-focused creative experience with an ideal balance of constraint and freedom: though the user now faces many more possible options at the beginning of the creative process, they can easily see which choices are cut off when they select a specific word for inclusion, and this prompts careful deliberation over which words are most essential to the intended meaning of the poem.

How could this experience of reflection be improved even further in the future? At present, the computational system in Redactionist is solely responsible for determining which words from an input text are selectable for inclusion in a poem, using a (somewhat flawed) part-of-speech tagging process. Adding a **reflective repair** step at the end of the poem creation process wherein *all* of the words in the input text become available for inclusion might help to preserve the initially helpful computational mediation of the creative process (which helps the user develop a stronger sense of what kind of poem they would like to create) while also allowing the user to deviate somewhat from the computer's idea of what constitutes a valid poem once their creative intent has been clarified.

Germinate

Germinate (Kreminski et al. 2020c) is a mixed-initiative CST for digital games that make arguments through procedural rhetoric (Bogost 2010). It presents the user with an interface for specifying the high-level rhetorical argument that they want to make through gameplay; a means of automatically transforming this argument into a variety of specific, playable digital games; and affordances for modifying the

[6] https://www.zachtronics.com/spacechem/



Design intent

View code

Game

Allow up to 2 entities

Allow up to 2 resources

Allow up to 6 triggers

Entities

Friend: Player thinks it's good

Resources

Depression: Player thinks it's bad; Value starts off low; Tends to increase slowly

Confidence: NOT Player thinks it's bad

Relationships

Friend produces Confidence

Triggers

When | Then

A Game about Depression

« Previous game 1 / 12 Next game »

Play

Depression Confidence

Restart game

Generated game rules

View code

Entities

Friend: Quantity several; Player thinks it's complicated

Stranger: Quantity several; Player thinks it's bad

Resources

Depression: Value starts off high; Player thinks it's bad

Confidence: Value starts off full

Relationships

Friend produces **Depression**

Friend produces **Confidence**

Stranger produces **Depression**

Stranger produces **Confidence**

Friend consumes **Stranger**

Triggers

When: Periodically: t_1_XX_

Then: Spawn entity: Stranger, 1

Figure 2: The Germinate user interface, with the user's current design intent on the left; a single generated game based on this intent in the middle; and design features of the generated game, which can be imported into the intent if the user likes them, on the right.

argument in response to these concrete realizations (Fig. 2).

Germinate already implements the three intent-related patterns (**elaborating**, **reifying** and **inferring intent**) described in this paper. It also implements **interpretive refraction** via the proceduralist readings of generated game rules that it conducts and surfaces to the user as a high-level summary of game dynamics. Additionally, Germinate facilitates **reflective repair** by allowing users to select specific rules and mechanics from flawed generated games and extract them directly into the design intent, enabling the preservation of these rules and mechanics in future generated games even as the rest of the design evolves.

However, Germinate's ability to facilitate reflection on how high-level design intents can be expressed via low-level game mechanics is limited by its current design. In particular, since the rules governing the system's understanding of how mechanics work together to create aesthetics are fixed and opaque to the user, the system may repeatedly attempt to realize a user's high-level intent through combinations of mechanics that do not actually support the intended player experience from the user's perspective. The system could more effectively support reflection on intent by opening these rules up to **reflective encoding**: perhaps first allowing the user to view the system's reasoning as to why it interprets a particular combination of mechanics as creating a particular target aesthetic, then letting the user disable interpretive rules that they disagree with or even introduce new ones as they develop their intent.

To support this process, an updated version of Germinate could implement **challenging choices** and **reflective revisitation** by occasionally prompting the user to annotate specific creative decisions with which aesthetic goals these decisions support. This might occasionally provoke users to realize that some of their choices do not effectively support the aesthetic goals that they are currently pursuing, maybe prompting a revision of the interpretive rules.

Additionally, as further support for **inferring intent**, a future version of Germinate could make use of discriminative learning for intent refinement. As in (Karth and Smith 2019) and (Kreminski, Wardrip-Fruin, and Mateas 2020), users could be asked to accept or reject generated games based on their alignment with the high-level design intent, and refinements of intent could be inferred from the shared characteristics of the accepted and rejected games.

Why Are We Like This?

Why Are We Like This? (*WAWLT*) (Kreminski et al. 2020b) is a playful, multi-user mixed-initiative CST for creative writing, powered by a social simulation engine that governs the behavior of a small cast of simulated characters. The system suggests actions for characters to perform, based on a model of character motivations and player-provided storytelling goals, and players choose which of these actions they would like to realize. Terse, system-generated descriptions of these character actions are then added to a running transcript of the story so far, which can be further annotated by the players with a more detailed description of each action.

In its current form, *WAWLT* provides support for **reifying intent** (by allowing users to specify explicit "author goals" for what they would like to happen next in the story) and **elaborating intent** (by suggesting actions that might fit the currently selected author goals). *WAWLT* also attempts to implement a limited form of **contextualizing choices**, both by showing each system-suggested action among several reasonable alternatives and by highlighting which author goals these suggested actions would immediately advance. This latter feature could also be viewed as a form of **interpretive refraction** in which the system evaluates prospective actions from the perspective of multiple distinct author goals at once.

A small-scale user evaluation of *WAWLT* (Kreminski et al. 2020a) found that some users struggle with a lack of clear direction when using the system, and sometimes forget to update their design goals as they move through the creative process. Taken together, these findings suggest that the system could support the progressive *refinement* of creative intent more effectively than it currently does. To that end, from a reflective creators perspective, we believe that *WAWLT* would benefit from a reworking of its interaction model to tie the reflective revision of intent more deeply into its core interaction loop.

By **inferring intent** on the basis of events that the user has already selected, a future version of *WAWLT* could more proactively identify and surface the goals that the users appear to be pursuing, and thereby prompt them to update their stated intent as their goals change. Additionally, an implementation of the **challenging choices** design pattern could help to provide users with a stronger sense of direction by making it clear to users when a proposed action *inhibits* one or more active author goals, or perhaps when the users have tunnel-visioned on the advancement of one author goal for several turns at the expense of others.

From a technical perspective, the implementation of *WAWLT*'s underlying generative system as a non-reversible simulation inhibits the ability to develop features that promote **reflective revisitation** and **reflective repair**. Past actions cannot be modified lest they implicitly invalidate future actions, so the system cannot prompt users to change the decisions contributing to their story's notional past. However, because the system allows users to write free text annotations for each event that has transpired, and because these annotations are not reasoned over by the system in any way, this may serve as an effective escape hatch for revisitation and repair: users can be prompted by the system to rewrite their descriptions of past events (perhaps to include some foreshadowing) if these events turn out to be pivotal later on in the story, or if they contribute to a high-level trend in the storyworld that is later reversed (such as two historically hostile characters eventually becoming friends, perhaps implying that the severity of past hostile interactions between these characters should be downplayed).

Conclusion

Not many reflective creators yet exist. However, it is our hope that by giving this category of systems a name and drawing together some design patterns demonstrated in existing examples, we will begin a conversation that leads to the development of more reflective creators going forward. The recent success of the "casual creators" label in drawing together practitioners and researchers with an interest in autotelic creativity support tools is inspiring to us in this regard, especially in light of the variety of work presented at the first Casual Creators Workshop at ICCC last year.

Many of the design patterns we have discussed in this paper add friction to the creative process. In our view, this is not necessarily a bad thing. Reflection is a viable process aesthetic in its own right, distinct from ease and pleasure; though approachability-focused CSTs may still want to prioritize ease and pleasure to eliminate barriers to entry, some users will always want to engage in reflection for the sake of reflection, even when it is challenging.

The development of reflective creators will not necessarily result in better *creative works*, especially short term. However, it might result in more thoughtful *creative practices* and practitioners.

Beyond ease, pleasure, and reflection, we believe that many other viable process aesthetics for autotelic creativity support tools remain to be discovered. By identifying one novel process aesthetic for CSTs, we hope to encourage other researchers to seek out further new aesthetics that are of interest to them.

Acknowledgements

Thanks to the Seabright Camerata, and especially Melanie Dickinson, for feedback on early drafts of this paper. Thanks also to the anonymous reviewers for their feedback, which led to substantial clarification of our argument.

References

Albaugh, L.; Hudson, S. E.; Yao, L.; and Devendorf, L. 2020. Investigating underdetermination through interactive computational handweaving. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, 1033–1046.

Alexander, C. 1977. *A Pattern Language: Towns, Buildings, Construction.* Oxford University Press.

Anthropy, A. 2012. *Rise of the Videogame Zinesters: How Freaks, Normals, Amateurs, Artists, Dreamers, Drop-Outs, Queers, Housewives, and People Like You Are Taking Back an Art Form.* Seven Stories Press.

Boehner, K.; Sengers, P.; and Warner, S. 2008. Interfaces with the ineffable: Meeting aesthetic experience on its own terms. *ACM Transactions on Computer-Human Interaction (TOCHI)* 15(3).

Bogost, I. 2010. *Persuasive Games: The Expressive Power of Videogames.* MIT Press.

Buchanan, B. G. 2001. Creativity at the metalevel: AAAI-2000 presidential address. *AI Magazine* 22(3).

Compton, K., and Mateas, M. 2015. Casual creators. In *International Conference on Computational Creativity*, 228–235.

Compton, K.; Kybartas, Q.; and Mateas, M. 2015. Tracery: an author-focused generative text tool. In *International Conference on Interactive Digital Storytelling*, 154–161. Springer.

Compton, K. 2019. *Casual Creators: Defining a Genre of Autotelic Creativity Support Systems.* Ph.D. Dissertation, University of California, Santa Cruz.

DeKerlegand, D.; Samuel, B.; and Leichman, J. 2020. Encoding socio-historical exegesis as social physics predicates. In *International Conference on the Foundations of Digital Games.*

Dickinson, M.; Wardrip-Fruin, N.; and Mateas, M. 2017. Social simulation for social justice. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 13.

Guzdial, M.; Liao, N.; Chen, J.; Chen, S.-Y.; Shah, S.; Shah, V.; Reno, J.; Smith, G.; and Riedl, M. O. 2019. Friend, collaborator, student, manager: How design of an AI-driven game level editor affects creators. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems.*

Hallnäs, L., and Redström, J. 2001. Slow technology–designing for reflection. *Personal and Ubiquitous Computing* 5(3):201–212.

Höök, K. 2008. Knowing, communication and experiencing through body and emotion. *IEEE Transactions on Learning Technologies* 1(4):248–259.

Hunicke, R.; LeBlanc, M.; and Zubek, R. 2004. MDA: A formal approach to game design and game research. In *Proceedings of the AAAI Workshop on Challenges in Game AI*, volume 4, 1722.

Isbister, K., and Höök, K. 2009. On being supple: in search of rigor without rigidity in meeting new design and evaluation challenges for HCI practitioners. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2233–2242.

Karth, I., and Smith, A. M. 2019. Addressing the fundamental tension of PCGML with discriminative learning. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*.

Krakauer, D. 2016. Will A.I. harm us? Better to ask how we'll reckon with our hybrid nature. https://nautil.us/blog/will-ai-harm-us-better-to-ask-how-well-reckon-with-our-hybrid-nature. Accessed on 2021-07-01.

Kreminski, M.; Dickinson, M.; Mateas, M.; and Wardrip-Fruin, N. 2020a. Why Are We Like This?: Exploring writing mechanics for an AI-augmented storytelling game. In *Proceedings of the Electronic Literature Organization Conference*.

Kreminski, M.; Dickinson, M.; Mateas, M.; and Wardrip-Fruin, N. 2020b. Why Are We Like This?: The AI architecture of a co-creative storytelling game. In *International Conference on the Foundations of Digital Games*.

Kreminski, M.; Dickinson, M.; Osborn, J.; Summerville, A.; Mateas, M.; and Wardrip-Fruin, N. 2020c. Germinate: A mixed-initiative casual creator for rhetorical games. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 16, 102–108.

Kreminski, M.; Karth, I.; and Wardrip-Fruin, N. 2019. Generators that read. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*.

Kreminski, M.; Wardrip-Fruin, N.; and Mateas, M. 2020. Toward example-driven program synthesis of story sifting patterns. In *Joint Proceedings of the AIIDE 2020 Workshops*.

Kybartas, Q.; Verbrugge, C.; and Lessard, J. 2020. Tension space analysis for emergent narrative. *IEEE Transactions on Games*.

Liapis, A.; Yannakakis, G. N.; Alexopoulos, C.; and Lopes, P. 2016. Can computers foster human users' creativity? Theory and praxis of mixed-initiative co-creativity. *Digital Culture & Education* 8(2):136–153.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2013. Sentient Sketchbook: Computer-aided game level authoring. In *Proceedings of the 8th Conference on the Foundations of Digital Games*, 213–220.

Löwgren, J., and Stolterman, E. 2004. *Thoughtful Interaction Design: A Design Perspective on Information Technology*. MIT Press.

Mahajan, S.; Bunyea, L.; Partlan, N.; Schout, D.; Harteveld, C.; Matuk, C.; Althoff, W.; Duke, T.; Sutherland, S.; and Smith, G. 2019. Toward automated critique for student-created interactive narrative projects. In *Proceedings of the AIIDE Workshop on Experimental AI in Games (EXAG)*.

Manjavacas, E.; Karsdorp, F.; Burtenshaw, B.; and Kestemont, M. 2017. Synthetic literature: writing science fiction in a co-creative process. In *Proc. Computational Creativity in Natural Language Generation (CC-NLG)*, 29–37.

Martens, C., and Hammer, M. A. 2017. Languages of play: towards semantic foundations for game interfaces. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*.

Nakakoji, K. 2006. Meanings of tools, support, and uses for creative design processes. In *International Design Research Symposium*, volume 6, 156–165.

Nardi, B. A. 1993. *A Small Matter of Programming: Perspectives on End User Computing*. MIT Press.

Nelson, M. J.; Gaudl, S. E.; Colton, S.; and Deterding, S. 2018. Curious users of casual creators. In *Proceedings of the 13th International Conference on the Foundations of Digital Games*.

Norman, D. A. 1991. Cognitive artifacts. In *Designing Interaction: Psychology at the Human-Computer Interface*. Cambridge University Press. 17–38.

Osborn, J.; Grow, A.; and Mateas, M. 2013. Modular computational critics for games. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 9.

Pérez y Pérez, R., and Sharples, M. 2001. MEXICA: A computer model of a cognitive account of creative writing. *Journal of Experimental & Theoretical Artificial Intelligence* 13(2):119–139.

Petrovskaya, E.; Deterding, C. S.; and Colton, S. 2020. Casual creators in the wild: A typology of commercial generative creativity support tools. In *ICCC'20: Eleventh International Conference on Computational Creativity*. Association for Computational Creativity (ACC).

Pousman, Z.; Romero, M.; Smith, A.; and Mateas, M. 2008. Living with Tableau Machine: a longitudinal investigation of a curious domestic intelligence. In *Proceedings of the 10th International Conference on Ubiquitous Computing*, 370–379.

Samuel, B.; Reed, A. A.; Maddaloni, P.; Mateas, M.; and Wardrip-Fruin, N. 2015. The Ensemble engine: Next-generation social physics. In *Proceedings of the Tenth International Conference on the Foundations of Digital Games (FDG 2015)*, 22–25.

Samuel, B.; Mateas, M.; and Wardrip-Fruin, N. 2016. The design of Writing Buddy: a mixed-initiative approach towards computational story collaboration. In *International Conference on Interactive Digital Storytelling*, 388–396. Springer.

Schön, D. A. 1983. *The Reflective Practitioner: How Professionals Think in Action*. Basic Books.

Sethapakdi, T.; Anderson, D.; Sy, A. R. C.; and Mueller, S. 2021. Fabricaide: Fabrication-aware design for 2d cutting machines. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*.

Shane, J. 2018. SkyKnit: When knitters teamed up with a neural network. https://aiweirdness.com/post/173096796277/skyknit-when-a-neural-network-teamed-up-with. Accessed on 2021-07-01.

Sharples, M. 1999. *How We Write: Writing as Creative Design*. Routledge.

Smith, A. M., and Mateas, M. 2011. Answer set programming for procedural content generation: A design space approach. *IEEE Transactions on Computational Intelligence and AI in Games* 3(3):187–200.

Smith, T.; Bowen, S. J.; Nissen, B.; Hook, J.; Verhoeven, A.; Bowers, J.; Wright, P.; and Olivier, P. 2015. Exploring gesture sonification to support reflective craft practice. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 67–76.

Smith, G. 2017. What do we value in procedural content generation? In *Proceedings of the 12th International Conference on the Foundations of Digital Games*.

Sullivan, A. 2020. Embroidered Ephemera: Crafting qualitative data physicalization designs from Twitter data. In *Joint Proceedings of the ICCC 2020 Workshops*.

Wiggins, G. A. 2006. Searching for computational creativity. *New Generation Computing* 24(3):209–222.

Zhang, A. X.; Hugh, G.; and Bernstein, M. S. 2020. PolicyKit: Building governance in online communities. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, 365–378.