Casual Creators

Kate Compton and Michael Mateas
Expressive Intelligence Studio
University of California, Santa Cruz
kcompton/mmateas@soe.ucsc.edu

Abstract

Many creativity tools exist to support task-focused creativity, but in recent years we have seen a flourishing of autotelic creativity tools, which privilege the enjoyable experience of explorative creativity over task-completion. Because these tools are much smaller in scope, less commercially significant, and less ”serious” than their larger siblings, they have been overlooked in academic research. This paper coins the term ”Casual Creators” for these tools, and provide a definition to identify tools that belong to this category. We also identify the particular design considerations that arise from autotelic creativity, and propose a number of strong design patterns that serve those considerations, patterns which are demonstrated by case studies of software built with those patterns. We believe that once this field is identified and named, the currently-isolated practitioners who make these casual creators will be able to share knowledge, like these design patterns, and develop a community of practice.

Introduction

An alternative design space

Often when we talk about tools to support creativity, the ‘creators’ exhibiting creativity are task-oriented professionals or amateurs, who have a specific problem to solve, or design task to accomplish. There exist many complex, powerful, and frequently expensive tools for different kinds of creative tasks: Maya for 3D modeling, FinalCutPro for editing video, Ableton for music production, Photoshop for images. These professional tools must support a broad range of possible actions with a focus on efficient task completion, as their users are typically being paid to complete predefined design tasks. Design, as an activity, is ”goal-oriented”, ”intentional, purposeful” (Gero 1994). ”Task” is an appropriate, and common, term for the primary action of using these tools, because the goal is to enable productive labor from the user.

Is this the only way to use creativity tools? Is productivity the only goal of creativity?

We propose the category of **Casual Creators** as an alternate design space for tools which support creativity as an intrinsically pleasurable activity, rather than as an extrinsically-motivated way to accomplish tasks.

Creativity is often an autotelic activity: we paint, draw, sculpt, sew, write and make music creatively, often with complete disregard for the quality of the final product (much less concern for task productivity), because the activity itself is so enjoyable.

This autotelic, intrinsically-rewarded form of creativity is psychologically quite distinct from creativity exhibited in a environment with extrinsic motivation (Amabile, Goldfarb, and Brackfleld 1990). We should expect that support tools for autotelic creativity will be correspondingly different. There is a thriving ecosystem of appropriately-designed software tools to support task-focused creativity, so why isn’t there a corresponding set of software to support autotelic creativity?

One reason is economic: the labor of the creative person is of commercial value to either their employer or their client, so there is a market incentive to maximize that output with effectively-designed tools, either to be purchased by the creative person themselves (for an independent contractor) or by their employer.

The other reason is the perceived “seriousness” of output. This division between serious and frivolous output mirrors the division in creativity research, between psychological (“P”) creativity and historical (“H”) creativity (Boden 2009). Psychological creativity is the ‘everyday’ (Kaufman and Beghetto 2009) creativity of average people, while historical creativity is the ‘eminent’ creative ability ascribed to famous world-changing creators and their innovative creations. Historical creativity is more valued socially and economically, so when building a tool to support creativity, tool designers like to imagine that it will be used by the next great artist or genius inventor (Shneiderman 2003), or at least used to make some famous or commercially successful product. We believe, however, that it is also important to build tools supporting ”everyday” creators in enjoying pleasant and fulfilling creative exercise, even if they never produce world-changing output.

Existing creators

Despite these reasons for not receiving ”serious” attention, there are many small applications that do exist to support casual creativity. The recently-developed app marketplaces for mobile have provided a perfect haven for this sort of creativity app. These apps each support creating only a single kind of artifact, such as abstract generative pictures (Secretan et al. 2011), virtual pottery ("Let's Create! Pottery" 2014), or 3d printable bracelets (System 2015). These apps create artifacts from a greatly-reduced possibility space, compared to the previously mentioned general-purpose professional tools. The narrowness of the possibility space allows the tool to provide greater support for the user, eliminating potential bad artifacts and speeding the process of creating good ones, at the expense of flexibility and versatility. This loss is acceptable, however, as these products aren't being created in response to the exacting demands of a boss or client, but rather because they are intrinsically *fun to make*. In fact, the end product may be discarded entirely after completion! (Nilsson 2003).

Other examples of autotelic creativity tools have come from games, a field that has always valued pleasurable user experience over productivity. Not every game contains creative activities, but many games do feature the creation (and curation) of a house, creature, or avatar, and these creativity tools can end up being more fun than the eventual gameplay.

And yet, though these tools exist, there is no central community in which their designers can communicate shared knowledge. There is no set of best practices that can be referenced by those who are attempting to make such a tool, or even a name for them so that the tool-maker can describe what they are making. The World of Warcraft (Blizzard 2004) character-creator and the "Let's Create! Pottery" virtual potter's wheel (to choose two out of many, many existing tools) may not seem to have much in common at first glance, or even share the same marketing category. We hope that by identifying this software genre, including lessons and design patterns learned from existing examples, we can define a distinct area for future research and tool implementation.

Introducing *Casual Creators*

Casual creators can be distinguished from other creativity support tools by their goal of supporting autotelic creativity, not task-focused creativity. From this initial difference in goals arises a variety of other differences: in design considerations, optimal design patterns, and the psychological states that they encourage in the user. To that end, we propose the following definition, which encapsulates the very exciting alternate design space of Casual Creators.

Definition

> A Casual Creator is an interactive system that encourages the fast, confident, and pleasurable exploration of a possibility space, resulting in the creation or discovery of surprising new artifacts that bring feelings of pride, ownership, and creativity to the users that make them.

Casual Creators are interactive systems. There are historical examples of non-digital casual creators, like the classic generative art toy Spirograph or the knitting toy Knit Magic, though digital software systems provide more affordances. They do, however, need to be interactive, driven by the user, because the learning and creating process is so core to the psychological experience of using one. Computational creativity can be used to assist the design process, but must be in a mixed initiative partnership with the user.

Casual Creators are tools that create artifacts, of some kind, which may be instances of virtual models or static images, or more abstract artifacts like story grammars or AI behaviors. Each creator has some *possibility space*, the set of all possible artifacts that could be created using that tool. The user creates (or discovers) artifacts by searching or exploring the space for 'good' artifacts. For the casual creator to be successful tool, there must be a way for users to find artifacts meeting functional and aesthetic criteria, avoiding getting stuck in a space of bad artifacts.

The possibility space should be narrow enough to exclude broken artifacts (such as models that fall over or break when 3D printed) but broad enough to contain surprising artifacts as well. The *surprising* quality of the artifacts motivates the user to explore the possibility space in search of new discoveries, a motivation which disappears if the space is too uniform. It also provides feelings of ownership and creativity when the artifact is discovered. In a sufficiently multidimensional possibility space, 'search' and 'creation' become blurred, as the only way to arrive at a particularly interesting artifact is to move through the space intentionally, rather than randomly searching. The user will feel greater ownership and creativity the more they attribute their discovery to their own actions, and pride is increased further when they feel that their discovered artifact is somehow special, *surprising* within the possibility space.

How does the user navigate this possibility space? Do they make tiny adjustments, tentatively inching through the possibility space, or do they make wild jumps, from solution to novel solution, exploring large regions of the space over a short period of time? An optimal creative process is described as making 'creative leaps', so we want to guide the user toward a fast-moving and confident exploration of the possibility space. The user's experience should feel playful, powerful, and pleasurable, like a flow state.

The user of a casual creator is a *casual* user, and the system can expect no previous domain knowledge, no previous technical experience, or adherence to a long learning process. All of the learning and creativity described above must occur in the first few minutes, and provide a good experience even if the user never spends time to gain mastery.

This definition focuses on the design goals of a Casual Creator, the experiences that Casual Creators are particularly suited to create. How we can design tools that *achieve* these goals is explained in the rest of this paper, through a description of design patterns and case studies that tested them. Some patterns come from existing Casual Creator-like tools, like the Spore Creature Creator, Nervous System's design tools (System 2015), and academic experiments like Picbreeder (Secretan et al. 2011). More patterns come from our current understanding of autotelic creativity, anticipating design patterns that support such a psychological state. To predict these new potential patterns, we draw from existing theories of creativity, flow, and design.

Related Fields

Reflection-in-action and Direct Manipulation

During the creative design process, the user modifies the artifact, moving quickly through a cycle of evaluation, planning, modification and reevaluation. This process shows the “Reflection-in-action” theory of learning, in which a learner hypothesizes, acts, and reflects on the results as a way to iteratively understand a domain or problem. The originator of that theory, Donald Schon, also applied it to the process of designing (Schon 1992) in which the designer ”sees, moves and sees again”. The *seeing* and *moving* are grounded in the materials themselves. This cycle cannot take place disembodied in the mind but must be enacted in dialogue with the artifact.

Seeing may include the user’s visual perception of the artifact, but is also a way to describe the evaluation of the artifact. Is it aesthetically pleasing, stable, strong? How does the designer predict that it will perform in its intended role? Some of these evaluations could be performed or assisted computationally. Schon surmises that while the reflective design process itself is not well suited to unsupervised computational processes, computation could provide new ways of “seeing”, or provide constrained micro design spaces “extending the designer’s ability to construct and explore them.” He concludes that “The design of design assistants is an approach that has not in the past attracted the best minds in AI. Perhaps the time has come when it can and should do so.”

In Direct Manipulation (Shneiderman 1993), a UI concept that parallels reflection-in-action, a complex software system provides ”continuous representation of the object of interest” and “[r]apid, incremental, reversible operations whose impact on the object of interest is immediately visible” and promises that “after obtaining reinforcing feedback from successful operation, users can gracefully expand their knowledge of features and gain fluency.”

The user can manipulate the system with rapid operations, then evaluate the effect immediately, because the artifact is always visible and responds immediately to the modification. The actions are reversible so the user is encouraged to experiment without anxiety, incremental, so subtle changes can be observed, and rapid (and rapidly seen), so that this learning cycle can operate continuously with each tiny iteration.

Flow

Csikszentmihalyi’s Flow theory is influential in games and creativity studies, but seems particularly well suited to the autotelic creativity of casual creators as“[i]deally, flow is the result of pure involvement, without any consideration about results.”(Csikszentmihalyi 2000) For flow to be achieved, the activity must have goals to create a sense of progress, immediate feedback so that progress can be sensed, and a balance between their perceived skills and challenges. Flow can be disrupted if the user feels frustrated, intimidated, or overwhelmed by choices.

Flow has a complex relationship with goals. Though the activity should be enjoyable in itself, without the pursuit of an outside reward, goals provide the required *direction and progress*. Goals can be provided as preset challenges, but often it is better to encourage the user to develop their own internal design goals. A good goal can be evaluated moment-to-moment, may change over time, and can be either highly specific to the user, or come from knowledge of the design space.

The flow state is very conducive to both creativity and an autotelic experience, and so provides important design considerations for potential Casual Creators, especially for avoiding conditions which disrupt flow, like choice paralysis or hard failures.

Creativity Support Tools

Lubart (Lubart 2005) identifies several categories of human and computer collaboration in the creative process: the computer can act as nanny, coach, pen-pal or colleague. Riedl and O’Neill (Riedl and ONeill 2009) suggest ”audience” as a fifth role for the computer. These categories provide a useful taxonomy, but do not provide implementable patterns.

The field of Creativity Support Tools, of which Casual Creators could be considered a subcategory focused on autotelic creativity, provides many concrete design patterns. Resnick et al (Resnick et al. 2005) identify many such patterns. Some, like “Support Exploration” and “Make It As Simple As Possible - and Maybe Even Simpler” are patterns to support flow experiences and reflection-in-action styles of learning. Other principles like “Support Many Paths and Many Styles”, “Low Threshold, High Ceiling, and Wide Walls” reflect how users will start with diverse goals and skills, which will further evolve as they use the system.

Some principles, “Choose Black Boxes Carefully,” “Support Collaboration,” and “Support Open Interchange”, ask the designer to reflect on the communities in which creative collaboration and learning occur, and how creativity develops as multiple users share knowledge. When we look at the creative communities that flourished for tools like Spore, Scratch (Resnick et al. 2009), and Twine (Klimas 2012), it becomes clear that, though the design of the single-user software is important, the technology decisions of data format, data interchange, hosting, and modifiability are equally critical to enabling creativity and fostering ownership. Creativity occurs between the user and client-side application, but also in the communities of practice that develops outside of the app, so creativity support tools must consider both sites, personal and communal (Maher 2012).

Generative Methods and Computational Creativity

Computational creativity is the science (and art) of encoding human-style creative process as automatable systems, with the goal of building a system which “exhibits behavior that would be deemed creative in humans.” (Colton et al. 2009). How ‘creativity’ can be detected in the finished artifacts of these systems is its own difficult problem (Maher 2012), but the field has successfully built generators that can design artifacts for domains as diverse as jokes (Petrovic and Matthews 2013) and paintings (Colton 2012).

These systems create artifacts by encoding the process of creating art (or literature, jokes, game levels, music, etc). The resulting algorithms must be able to create not only one successful example, but a wide and interesting space of possible valid artifacts, some of which should be able to surprise even the person who wrote it. Such algorithms can be called *generative methods* (Compton, Osborn, and Mateas 2013); they use a range of technologies (genetic algorithms, grammars, declarative modeling), but all share the goal of creating large possibility spaces of valid-yet-surprising artifacts. This is the optimal type of possibility space for computational creativity systems, and also for Casual Creators.

Computational creativity and generative methods are often a poor fit for productivity-focused creativity apps. Professional creativity involves creating to very specific requirements, requires complete control and the ability to fine tune the resulting work. Generative methods create a lot of work, very fast, but with minimal control over the output (compared to hand-authored content) and often no way to iterate on the output. A casual user, *without* the need for complete control, is willing to trade a loss of control for the speed, power, and surprise of generative methods.

The expressive range of such systems must always be balanced with the need to produce valid content. A system could produce a wide variety of mostly broken artifacts, or produce a set of high-quality yet homogeneous artifacts, but both of these are failures. We have found the phrase '1000 bowls of oatmeal' useful to describe the common antipattern of generating a set of artifacts which are technically distinct to the computer, but perceived by humans as uniform.

Computational creativity systems usually run autonomously and unsupervised by humans. Pairing these methods with human users can add additional power to the process (Davis et al. 2014), as humans provide aesthetic evaluations and intuitive leaps to the rapid generativity of the computation creativity processes. Mixed initiative systems, in which the computer and human users operate simultaneously or by turn-taking, support a creative cycle in which each user reflects on the previous contributions of their collaborator and modifies the artifact according to their particular abilities. The end products of the creative process are improved, and ideally the user enjoys the experience of collaboration, if the system is well designed. Interaction with a highly generative system has a particular set of pleasures, whether in the context of a game or a creativity tool. Chaim Gingold refers to such pleasurably interactive systems as 'Magic Crayons' (Gingold 2003): computational, accessible, sketchable, expressive systems which invite the user to play with them and discover hidden secrets and affordances.

Design Patterns

The definition of a Casual Creator as an autotelic creativity tool provides an abstract guide for what we would want a potential Casual Creator to accomplish. To actually design such a tool, these high-level patterns must be interpreted into concrete design patterns. We have identified a number of these patterns, drawing from existing Casual Creators, and from the related fields, and tested them by using them to create a wide variety of systems, described in the Case Studies section below. These design patterns are not exhaustive, but are representative ones that are versatile, common, and easy to apply.

Instant feedback Recall that both direct manipulation and reflection-in-action require the user to observe the artifact, make a change, and see the results, a process which allows them to discover patterns and affordances in their possible changes, mastering the system while iterating on an artifact. In the *instant feedback* pattern, the changes should be immediately visible in the modified artifact. However, just visually regenerating the artifact in response to changes, even in real-time, is not necessarily enough to provide appropriate feedback.

'Seeing', in the reflection-in-action model, encompasses more than just 'looking at'. 'Seeing' actually encompasses the entire process of sensing and evaluating the artifact's fitness according to both the potential use case and the user's own design model. For objects with a strictly aesthetic role, this is easy: the user glances at it, and can instantly decide their opinion of it. Other evaluations are complex, and must be either mentally simulated by the user, or else evaluated by the system. Requiring the user to mentally simulate complex consequences will take a lot of time and attention, and the evaluation could be inaccurate or flawed, slowing the iteration process. The *instant feedback* pattern would recommend computationally simulating and visualizing as much as possible so that the user can get feedback at a glance.

The Chorus Line Named after the choreography concept in which many dancers all execute the same routine simultaneously, the *chorus line* pattern was used in Spore (Hecker et al. 2008) as an internal tool to test animations on a wide range of creature morphologies. The chorus line is a subpattern for *instant feedback*, in situations where what is being generated is not a single artifact, but a space of artifacts. In that situation, the user should be able to 'see' (in the reflection-in-action sense) the space of their creation, instantly. Instead of generating one example, this pattern suggests generating many examples, and overlaying them (spatially, temporally, graphically) to make subtle differences and similarities easier to spot.

Simulation and approximating feedback Automated visualization becomes especially important when the artifact being generated would take minutes or even hours for the user to evaluate, rather than milliseconds for an image, or seconds for an animation. For artifacts such as game levels, the artifact is judged by the many gameplay traces over time that could be played on it, which cannot be visually evaluated with much accuracy by a casual user. Nor can a system show the user all possible gameplay traces, so the user must be shown a proxy of the evaluation. When Riedl and O'Neill (Riedl and ONeill 2009) add 'computer as audience' to Lubart's categories, their simulation proposed to accurately model how a human reader would evaluate generated stories. In Sentient Sketchbook (Yannakakis, Liapis, and Alexopoulos 2014), the system calculates "navigational and topological properties" as the user interacts with it, providing instant feedback for a complex artifact. This evaluation does not fully encapsulate the actual gameplay implications of the map, which for a finished level being put in a game, could be a potential design issue. However, for a Casual Creator, the goal of the evaluation is to provide the sense of progress towards a goal necessary for achieving flow. Only the *perception* of progress is necessary: as long as the user perceives progress, the accuracy of the evaluation is irrelevant.

Entertaining evaluations One nice benefit of relaxing the need for accurate evaluations is that the evaluations can *themselves* be pleasurable and entertaining. In the Spore Creature Creator, when the user modifies their creature the creature will respond by laughing and shaking the new body part in appreciation, or, less commonly, expressing distress. The choice of happy or sad reaction does not actually represent any real system state, it just provides arbitrary feedback. That feedback is psychologically significant, for encouraging the flow state, but also for letting the user feel pride in pleasing their little AI judges. Even if the user starts with no particular design direction of their own (a common issue with casual artists) having a simulated critic present can suggest a direction for the user, even if they choose to ignore it. The abstract generative art game BECOME A GREAT ARTIST IN JUST 10 SECONDS (Brough and McClure, 2014) waggishly compares the user's glitch art to classic masterpieces and rates it with a percent similarity, an intentionally arbitrary metric that still serves to provide optional direction to the casual user.

No blank canvas One benefit of focusing on these intrinsically-motivated users is that they are often much more flexible about the final product. In contrast to a system like Maya, which must support extremely broad use cases and a high degree of fine-tuning in order to make a very particular finished product, a casual user will have more flexible requirements for their product. They likely want it to be functional and aesthetically pleasing, but are willing to consider many more possible kinds of solutions, or may not even start with any particular solution in mind (Nilsson 2003)

Professional artists know the terror that comes from facing a blank canvas (Bayles, Orland, and Morey 2012), but this experience is also intimidating and paralyzing to the novice user. However, this can be very easily mitigated, by providing either a starting shape (Spore) or a suggested challenge (Let's Create! Pottery). The first move is the hardest, so this restricts the first move to a single decision: accept the prompt, or discard it. Once this one move is taken, subsequent actions are easier.

Limiting actions to encourage exploration This restriction of actions can be useful even after the user has moved past the blank-canvas moment. To achieve a flow state (Csikszentmihalyi 2000), the user should be able to quickly and confidently make decisions, which is easier if the available choices are appropriate, limited in number, and their consequences are clear (or at least suggested) to the user.

One strategy is modal interaction: limiting actions by the particular mode that the creator is in. This approach is common in character creators like that in World of Warcraft and Spore Creature Creator, which have different modes corresponding to user actions like painting or building and panels with sub-actions within those modes, to choose hair or faces, revealing only actions for the mode that the user is currently in. Another approach is to limit the actions available, slowly unlocking them in response to experience, challenges defeated, purchases, or some other pacing mechanism. If the possibility space is temporarily restricted, the ability to more fully explore the space scaffolds the user's understanding of the possibility space.

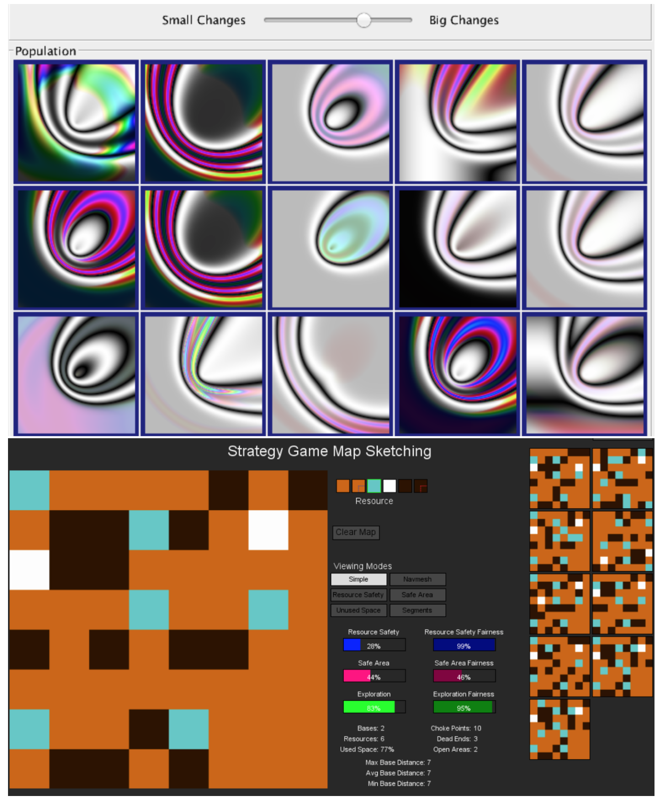

Figure 1: Top: *Mutant-shopping* for images in Picbreeder. Though the user can control the rate of mutation, they can only 'create' an image by selecting the parents of the next generation. Below: Sentient Sketchbook shows automated evaluations, allows direct editing, and also provides some alternate mutants on the right

Mutant shopping One feature that can help the user find unexpected solutions in the possibility space is not a creative 'action' at all, but the availability of suggested alternatives, like artifacts near the current one in the possibility space. In some tools, the user is not given any way to edit the artifact, and must navigate the possibility space by picking one of the new options, as in Picbreeder (Secretan et al. 2011).

In other cases, as in the parametric tree modeler Dryad (Talton et al. 2008), the user may use these alternative to browse the space, but can also further edit the artifacts that they discover in that way. A third framing of this pattern is found in Sentient Sketchbook (Yannakakis, Liapis, and

Alexopoulos 2014), in which the user edits a game level normally, but the system uses that information to generate additional suggested artifacts that are 'nearby' for some more abstract calculated metric, rather than 'nearby' in their underlying representation.

Although this process has a lot in common with evolutionary algorithms (specifically human-guided evolutionary algorithms (Klau et al. 2010)), the focus is not on producing an optimal specimen, but on the enjoyment that the user feels from this process. For this reason, we named this pattern *mutant shopping* to capture the psychological pleasures and motivations of a less-directed browsing and discovery process like shopping.

Modifying the meaningful In Spore, parts can be placed anywhere on the creatures, then modified by rotation or pulling on their morph handles. In a traditional sculpting program like Maya, these handles would be expected to control a clear parameter like z-scaling, for maximum control over the changes. The Spore designers discovered that it was more interesting to have these handles control higher level changes, like shifting a jaw from top-heavy overbite to jowly underbite, or extending a foot's shape from round toes to pointed claws. Higher-level modifications like these give the user a more meaningful space to explore.

Saving and sharing As noted in the "Design principles for tools to support creative thinking" report, the client-side application where the user is editing their artifact is only one site where creativity occurs, and designers of Casual Creators should also consider how they support creativity *outside* of their app. One example of this principle is the use of common, free, human-readable filetypes for saving data, such as JSON or images. Spore embedded the creature's save data stenographically in a PNG image, and the latest version of Twine 2 embeds the editable hypertext into the HTML that plays the Twine game. Even if the client app is still necessary to rebuild the content from the saved data, as in Spore and Twine, users can share their data using existing platforms. Most hosting sites allow text and images, but not arbitrary files. If users can easily host their save files on such hosting sites, they can build communities independently from the makers of the original casual creator app.

Hosted communities An alternate pattern is to provide a hosted community that is tied more closely to the client app, as Picbreeder and Let's Create! Pottery do. Casual creators should encourage the user's pride in their discovered or created artifacts, so providing a showcase where user's can publish their work to share it with others supports this feeling of ownership. Creations are often annotated or tagged, and usually there is a commenting and messaging system, enabling a large community to communicate within itself. Modification is its own form of communication, so if the system supports modification of artifacts, they should show their ancestry, and notify the original creators so that they can take pride in their influence.

Modding, hacking, teaching Users of casual creators will quickly find that the tool does not support every action that they want. The tool and its surrounding community support should facilitate users in teaching each other mods and hacks that expand the boundaries of what's possible with the tool. The previous two patterns support this pattern, as this teaching can happen on external sites, or internal ones, but the easier it is to find a clever hack, import it into the tool, and modify it and republish the new results, the quicker these ideas will spread through the community.

Case Studies

Instant Feedback: PendantMaker

PendantMaker is an online design tool for creating 3D printable pendants. We observed that although 3D printing is interesting to many people, the tools to create printable content are difficult, with many potential pitfalls for making unprintable and broken content. By restricting the domain space to extruded tubes, we could guarantee that our generated geometry would be valid for printing, and print reliably on a cheap printer (a difficult set of physical constraints). When combined with turning sliders, supporting the Direct Manipulation patterns of "rapid, incremental, reversible operations" (Shneiderman 1993), PendantMaker provided a very 'safe' place for the user to experiment without fear of failure.

We also noted that casual users often doubt their drawing ability (Bayles, Orland, and Morey 2012) and lack direction, so we designed a generative algorithm in which undirected scribbles from the user would be reflected around an axis, creating a design of surprisingly attractive symmetry. We provide a canvas for the user to draw a line, which is extruded, shaped, and reflected into the many intersecting tubes on the right, creating the printable pendant *in real time*. This very immediate feedback was critical: users could draw aimlessly, but notice when the reflections would intersect or join together, allowing the users to easily create a complicated knotwork of intersecting tubes that would be impossible to predict without feedback. We also added sliders for a variety of tuning values, reflecting the *Modifying the Meaningful* pattern above. Some sliders corresponded to clear values like thickness and arm count, but 'bloom' performed a complicated sculptural task of flaring the outermost tubes in a curved shape. Complicated tools like bloom are only usable with rapid feedback: their action is indescribable to the user, but with a little experimentation, the user quickly learns how to use them artistically.

Sharing and Ownership: IceMaker

IceMaker was an evolution of PendantMaker's design, to create extruded 3D snowflakes, and similarly uses tuning sliders, symmetry, and extrusion to create complex geometry that is both modifiable and guaranteed valid, with immediate feedback. The extrusion path is not controlled by the user's drawing, as in PendantMaker, but rather by a particle simulation. The behavior of the particles would be *very* hard for a casual user to program, so instead we provide sliders for values that represent the resulting appearance of the path ('complexity', 'wiggle', 'sharpness') allowing the user to explore the possibility space while not having to understand the complex process behind it (Fig. 2).

Figure 2: Ice-Maker, a 3D snowflake maker, guides the user to create a snowflake and further personalize it with a message, then embed the design into a single URL that the user could share.

Since this interaction provides less agency than the drawing interface in PendantMaker, we wanted to augment Ice-Maker with other ways to declare ownership of the discovered snowflake. Following the *Saving and Sharing* pattern, we encoded the snowflake into a unique URL which the user could share, post or send as a 'saved' version of their artifact.

Search and Discovery: Funky Ikebana and Tiny Dancer

Creativity-as-discovery is further explored in Funky Ikebana, in which L-system flowers are are generated from a 'DNA' of floating point tuning values. Similar to exploration process in Dryad (Talton et al. 2008), the user iteratively selects the flowers that they like, and the system generates more nearby examples. This human-guided evolutionary algorithm allows for 'optimization' of the flower, but as this was designed as a Casual Creator, we focus on the pleasures of *mutant shopping* more than potential optimization. The flowers are arranged together, which makes different ones easy to spot, so the user can pick from flowers that are very different, or mostly the same. Regeneration when one flower is selected and its children repopulate the space is instantaneous, so the user can very quickly move through the space of flowers. Picking from one of 10 children limits the number of actions that the user can take, so choice paralysis does not occur, as shown in previous mutant shopping examples like Picbreeder (Secretan et al. 2011).

Unlike the Dryad and Picbreeder systems, we were also able to use the L-system to create a simple animation for the flowers, causing them to 'dance'. Flowers danced differently, an emergent property of their morphology, and users could selectively evolve flowers for their movement instead of just shape. Because we used the *chorus line* pattern to show many flowers dancing at once, the user was able to notice particularly graceful or vigorous ones, and select for that. Tiny Dancer takes this idea further by simultaneously evolving the morphologies of ragdoll dancers and their dance-responses to music, so that the dances can also be selected by mutant shopping on a chorus line.

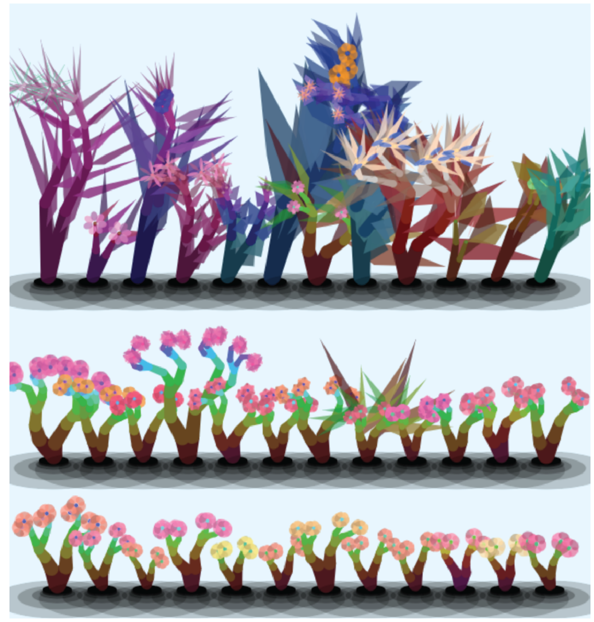

Figure 3: Iteratively evolving smaller flowers in Funky Ikebana, starting with the center flower in the first image. The user's current heuristic is to pick small simple flowers, but that heuristic can change each time the user spots a flower style that they like better.

Interventions: BotPrint and Binary Fission

The Casual Creator framework has been usefully applied as an intervention in two existing designs, successfully modifying the designs to improve the user's creative experience.

BotPrint was an existing application to design laser-cuttable robotics kits for children. Users could drag handles to shape the outline of the bot's chassis, and some automation would occur to figure out placement of components. Unfortunately, the implications of moving components and changing chassis size were not visible to the users, so making modifications felt meaningless. Using casual creator design patterns, we updated the system to simulate the bots moving in an 'arena' with many other similar bots. This provided a way for the user to evaluate the behavior of the bots visually (chorus line), see and select variants (mutant shopping) and enjoy watching the bots struggle for victory (entertaining evaluations), while also directly modifying the bots and then rereleasing them into the arena.

Binary Fission is a game designed to help the user make binary decision trees to filter loop invariant data for a crowd-sourced science task about software verification. At first, this does not seem to be a creative task, but by using casual creator patterns to emphasize the creative side of selecting the filters to build the filter tree, users enjoyed the task and were able to explore the possibility space of trees much faster. The biggest insight provided though the casual creator lens was to show many filters for each choice point. Calculating how well a filter would behave is a very arduous evaluation for the user to perform themselves, so we colored each by how well it filtered data at that point. Users were able to glance through this potential ‘filter space’ for suitable filters, and were able to apply them, see their implications, and rebuild trees very quickly, turning what could have been an opaque and arduous task into a fun reflection-in-action learning experience.

Conclusion

This paper defines a new term, *Casual Creators*, to identify a category of interactive systems which prioritizes the experience of autotelic creativity above productive output, an exciting new design space that is distinct from existing productivity-focused creativity support tools. We have illustrated the distinct design considerations of *Casual Creators* by identifying and describing representative design patterns drawn from theories of creativity and current successful systems. These patterns were used to design several new systems, and to evolve some existing designs to better support casual creativity. From these case studies, we learned that these patterns do clarify and inspire the process of building systems to support casual creativity, as it was easy to identify new system features from the patterns. Additionally, using the lens of *Casual Creators* enabled us to easily find examples of those features in a wider range of existing systems than would have otherwise been possible.

References

Amabile, T. M.; Goldfarb, P.; and Brackfleld, S. C. 1990. Social influences on creativity: Evaluation, coaction, and surveillance. *Creativity research journal* 3(1):6–21.

Bayles, D.; Orland, T.; and Morey, A. 2012. *Art & fear*. Tantor Media, Incorporated.

Boden, M. A. 2009. Computer models of creativity. *AI Magazine* 30(3):23.

Colton, S.; López de Mantaras, R.; Stock, O.; et al. 2009. Computational creativity: Coming of age. *AI Magazine* 30(3):11–14.

Colton, S. 2012. The painting fool: Stories from building an automated painter. In *Computers and creativity*. Springer. 3–38.

Compton, K.; Osborn, J. C.; and Mateas, M. 2013. Generative methods. In *The Fourth Procedural Content Generation in Games workshop, PCG*.

Csikszentmihalyi, M. 2000. *Beyond boredom and anxiety.* Jossey-Bass.

Davis, N.; Popova, Y.; Sysoev, I.; Hsiao, C.-P.; Zhang, D.; and Magerko, B. 2014. Building artistic computer colleagues with an enactive model of creativity. In *Proceedings of the 5th International Conference on Computational Creativity*.

Gero, J. S. 1994. Towards a model of exploration in computer-aided design. In *Formal design methods for CAD*, 315–336.

Gingold, C. 2003. *Miniature gardens & magic crayons: Games, spaces, & worlds*. Ph.D. Dissertation, Georgia Institute of Technology.

Hecker, C.; Raabe, B.; Enslow, R. W.; DeWeese, J.; Maynard, J.; and van Prooijen, K. 2008. Real-time motion retargeting to highly varied user-created morphologies. In *ACM Transactions on Graphics (TOG)*, volume 27, 27. ACM.

Kaufman, J. C., and Beghetto, R. A. 2009. Beyond big and little: The four c model of creativity. *Review of general psychology* 13(1):1.

Klau, G. W.; Lesh, N.; Marks, J.; and Mitzenmacher, M. 2010. Human-guided search. *Journal of Heuristics* 16(3):289–310.

Klimas, C. 2012. Twine. Accessed: 2015-02-24.

Lubart, T. 2005. How can computers be partners in the creative process: classification and commentary on the special issue. *International Journal of Human-Computer Studies* 63(4):365–369.

Maher, M. L. 2012. Computational and collective creativity: who’s being creative? In *Proceedings of the 3rd International Conference on Computational Creativity*, 67–71.

Nilsson, B. 2003. ‘i can always make another one!’–young musicians creating music with digital tools. *Musicianship in the 21st century: issues, trends and possibilities (Sydney, Australian Music Centre)*.

Petrovic, S., and Matthews, D. 2013. Unsupervised joke generation from big data. In *ACL (2)*, 228–232.

Resnick, M.; Myers, B.; Nakakoji, K.; Shneiderman, B.; Pausch, R.; Selker, T.; and Eisenberg, M. 2005. Design principles for tools to support creative thinking.

Resnick, M.; Maloney, J.; Monroy-Hernández, A.; Rusk, N.; Eastmond, E.; Brennan, K.; Millner, A.; Rosenbaum, E.; Silver, J.; Silverman, B.; et al. 2009. Scratch: programming for all. *Communications of the ACM* 52(11):60–67.

Riedl, M. O., and ONeill, B. 2009. Computer as audience: A strategy for artificial intelligence support of human creativity. In *Proc. CHI Workshop of Computational Creativity Support*.

Schon, D. A. 1992. Designing as reflective conversation with the materials of a design situation. *Research in Engineering Design* 3(3):131–147.

Secretan, J.; Beato, N.; D’Ambrosio, D. B.; Rodriguez, A.; Campbell, A.; Folsom-Kovarik, J. T.; and Stanley, K. O. 2011. Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary Computation* 19(3):373–403.

Shneiderman, B. 1993. Direct manipulation: a step beyond programming languages. *Sparks of innovation in human-computer interaction* 17:1993.

Shneiderman, B. 2003. *Leonardo’s laptop: human needs and the new computing technologies*. Mit Press.

System, N. 2015. Nervous system.

Talton, J.; Gibson, D.; Hanrahan, P.; and Koltun, V. 2008. Collaborative mapping of a parametric design space. Technical report, Citeseer.

Yannakakis, G. N.; Liapis, A.; and Alexopoulos, C. 2014. Mixed-initiative cocreativity. In *Proceedings of the 9th Conference on the Foundations of Digital Games*.