MANIPULUS

Ian Bellomy
2012

Abstract

Interaction is experiential but also computational. Creating interactive artifacts requires programming which is difficult and time consuming. Programming also requires a clear understanding of the artifact's desired behavioral qualities. Unfortunately, the qualities of interaction are nebulous in general, and the interactive qualities of a specific concept may be inaccurate, malformed, or simply ineffective in engendering the envisioned experience. Better understanding is predicated on an ability to quickly create and experience different interactive possibilities, but this is curtailed by the difficulty of programming. The problem is circular. Ambiguities in our understanding of interaction make programming more difficult which in turn obscures understanding of interaction.

This thesis address the issue by proposing a model of single user, screen based, interaction that includes low level qualities both perceptually descriptive and computationally tenable based on defining interaction as an emergent quality of manipulation. A series of simple, non figurative, interactive, studies is created to test where the model may be deficient, ambiguous, or computationally problematic by probing potential boundaries between manipulation and interaction. Though the model is device agnostic, investigations are limited to typical mouse or trackpad input. Lastly, an implementation of the model in Actionscript 3 is also included along with a rudimentary interface prototype that allows for the direct manipulation of a limited set of the proposed qualities.

Introduction

1.1 Ambiguity

As an ongoing area of research, the study of interaction or interactivity comes with a variety of definitions and overlapping usage of the terms interactive, interactivity and interaction for reasons both historical, perceptual, and colloquial. Dag Saevanes provides the most clarity on the general definitions of these terms; *Interaction* refers to the reciprocal action between a person and a computational device. (This thesis focus specifically on single user screen based systems. I'll further restrict the usage of this term to refer to the user's subjective experience of interaction.) *Interactive* is an adjective referring to an artifact's ability to support interaction. Its *interactive qualities* being those qualities — whatever they may be — that make it distinct from other visual and kinetic artifacts. (The specific nature of these qualities are relevant to criticism, analysis, design, and tool creation and are an active area of research in fields including Human Computer Interaction, Interaction Design, Interactive Design, and Game Design[1].) Lastly, *interactivity,* noun, is the general phenomena that unites interactive artifacts.

Interactivity creates particular problems with traditional iterative design practices. In traditional design fields, potentially nebulous initial concepts are quickly explored to find the most fruitful direction for investigation. The specifics that contribute to the gestalt of traditional form (point, line, plane, texture, volume, etc.) are relatively well understood, and isolating the most effective combination of these qualities to provide a generalized 'outline' of a potential solution is central to initial exploration. Unfortunately, interactive qualities that contribute to the character of an experience may not be observable in traditional sketches or layouts. Interactive artifacts, while predominately visual and kinetic, often posses qualities that are only apparent during a direct and active encounter (Löwgren and Stolterman) and may remain hidden until an interactive version of the artifact can be encountered. This would be less problematic if the actual creation of functioning prototypes was less difficult.

Ware, New Basics, Elements of Design

1.2 Programming

Addressing the difficulty of creating prototypes is a common and useful approach. The general idea is that reducing programming difficulty or improving programming knowledge leads to more and/or faster exploration and from this more experience and understanding of interaction and interactivity.

An interactive artifact in the most technical sense is a continuous, interruptible, process running on a computer that controls the creation of visual forms that vary in response to a context outside of the artifact. Programming languages provide the precise detail needed to annotate these processes in a manner a computer can execute. While often an immutable and essential task in creating the artifact, creating and modifying these programs to produce desirable and innovative[2] results is both difficult and time consuming (Buxton 97).

Some methodologies aim to separate and isolate programming from the 'design task at large' in order to minimize the difficulties in programming — specifically to avoid a lack of conceptual integrity and correspondingly ambiguous and costly implementation needs (Brooks 42). These methodologies emphasizes a view of programming as the production work of a fully formed idea (Reas et al. 25). The aim is to minimize the risk of complicated development issues by strictly defining the needs of the program. This was the very explicit goal of the 'waterfall' development methodology (Sharp ??) and now, more loosely, a complimentary goal of user centered design: Programming is difficult and time consuming, do it as late as possible (Sharp et al. ??). In game design instruction, the programming problem is sometimes avoided by emphasizing the creation of board games over digital games — in these cases a student/designer must still create an algorithmic system but with the leniency that it need only be executed by people opposed to computers. This phenomena is mirrored in the digital product realm through the use of paper prototyping techniques.

While pragmatic in concept, this approach is at odds with traditional iterative practices found in graphic and industrial design. (The approach is also problematic from a development standpoint as the creation of software often involves the uncovering of new problems unaddressed in initial specifications (Brooks 264)). Iterative practices, as Colin Ware describes, emphasize a designer's repeated encounters with evolving manifestations of their ideas; each iteration modified in order to balance the complex requirements of audience, subject material, and medium. While the visual qualities of an interactive artifact may be easily prototyped using the designer's intuitive visual problem solving abilities and drawing skills, rapid prototyping — or sketching — of the interactive qualities of these visual forms is reliant on a designer's programming skills and, more importantly, their ability to identify and abstract potentially complex behaviors that may be central to the artifact (Gingold, "Catastrophic Prototyping").

Promoting the development of programming skills is another approach. Attempts at helping students overcome conceptual hurtles inherent in programming is at least as old as the LOGO programming language created in 1967. Since then there's been numerous attempts to make programming more accessible (particularly for artists and designers) including software like Hypercard, Director, and MaxMSP, languages (or programming libraries) like Lingo, Processing, and Open Frameworks, and learning aids like codeacademy.com. Each of these, while achieving a variety of successes and support from communities, re-encounter similar difficulties. Alan Kay in his foreword to *Watch What I Do*, explains how even simple scripting languages represent a less than ideal learning investment. "1) Users still have to learn the arcane syntax and vocabulary conventions of the language, and 2) they have to learn the standard computer science concepts of variables, loops, and conditionals." Brooks explains how the difficulty in programming may be intrinsic. "Because a programmer builds in pure thought stuff, we expect few difficulties with implementation. But, our ideas themselves are faulty, so we have bugs.” *(pp)*

page

While strict in syntax, programming languages are highly expressive. Their abstraction allows writing processes that read input and control output in a variety of contexts completely unrelated to the visual and interactive problems a designer is required to deal with — such as cryptography, database management, or networked communication. Languages like *Processing* mitigate this by hiding instructions unessential to a designer or artist while providing instructions with more immediate visual consequences. This allows designers to affect screen visuals more quickly and restores some of the continuity of the iterative process. The *Processing* community even refers to their programs specifically as ‘sketches’. This type of approach is invaluable for both full projects and in educational contexts.

More ideal though would be a programming language that provides a brief but powerful instruction set for affecting a form’s interactive qualities, so that perceived and imagined behaviors could be more easily translated into a computational form and vice versa. As form and interactivity are inseparable (Saevens) such an isolated language may be impossible. However, a better understanding of the subjective experience of interaction may point to a more ideal vocabulary. ~~To be useful, such a model must be quantifiable in order to support computation and also describe the qualitative aspects of the subjective experience of interaction.~~

erp, pp

1.3 The encounter

Interactive artifacts, while predominately visual and kinetic, often posses qualities that are only apparent during a direct and active encounter. This is what Löwgren and Stolterman describe as the ‘dynamic gestalt’. While an artifact may be visual and kinetic, its interactivity — its interactive qualities — affect how an individual perceives the artifact as a whole, just as color interplays with form to support an static image’s gestalt. The results of this interplay are, of course, variable.

Firstly, there is nothing to say that an active encounter necessarily makes for a “better” experience or that interactivity is a panacea for an otherwise dull or ineffectual design. Active encounters may even be less enjoyable than passive ones. Conversely, modifications to visual qualities may improve the perception of an artifact’s interactivity.[3] Regardless of the quality of the experience, the simultaneous interplay of an artifact’s visual, kinetic, and interactive qualities typically engenders a more comprehensive understanding.

It’s also possible for an interactive artifact’s overall perception, meaning, or significance to be dominated by its visual, kinetic, or narrative qualities. The analysis of games like *Uncharted : Drake’s Fortune* that rely heavily on story and characterization would be appropriately evaluated as a linear art form as much as an interactive one.

Some, or even many aspects of an interactive artifact may even be understood through observation (Lopes 101). Viewing an interface in use may provide a wealth of information about how it may be used and may shape the experience of a future encounter. Design methodologies utilizing scenarios or story boards may provide insights into the desired experience in relatively short order[4]. Similarly, some interactive artifacts may even be “understood” through simple contemplation of use. Löwgren explains:

> *“Parafunctional design is generally appreciated in three steps, starting with a simple recognition of the product and its intended function, followed by a brief period of frustration at the obvious inappropriateness of the intended function and only then a sudden insight (the »a-ha« moment) when you realize what the artist-designer wants to make you see.”*

example

Lastly, an active encounter may fail to reveal the full nature of the artifact. Some require prolonged or reoccurring use in order to completely understand their qualities (such as an air traffic control system (Dourish)). Similarly, some artifacts may influence behavior in ways that are not understood to the users themselves (or the public) and require reflective and critical evaluation if not empirical studies. The effects of social networking sites or violent video games on child social and mental development are particular areas of interest.

page

Outside of these exceptions, interactive artifacts are generally qualified by the value of a direct encounter. The existence of game*play* underscores this phenomena. Games typically have no direct utilitarian use, but may provide a more elaborate, rich, detailed, or full experience when encountered. As a subset of interactive artifacts, they must deal with issues the phenomena along with their own domain specific issues. Issues of ambiguity are echoed in game design design writings repeatedly; Game designer Doug Church voiced an open request for an abstract formal language. The lack of a coherent model drove Greg Costikyan to write *I Have No Words and I Must Design*. Jesse Schnell notes that in the absence of clear definitions forces designers to make do with a variety of lenses (24).

While the existence and value of an interactivity is clear, its specifics are not. To my knowledge, at the time of writing[5] there is no broadly accepted model of interaction with the low level detail comparable to that found in traditional design languages concerning gestalt-forming elements. Whether interactivity can be dissected into discreet dimensions as color is dissected into hue, value, and saturation, or form into point, line, and plane is unclear. However there is a wealth of writing on the topic at large, some of which serving as a direct foundation for this project’s investigation.

Firstly, it is widely held that basic perceptual and cognitive abilities are biologically contingent and thus relatively consistent (Norman, Raskin, Ware). As such, a model of interaction based on potentially universal similarities in the perception of interaction should at least be possible.

While the experience of interaction may be contingent on factors such as experience, age, and culture, this project will focus on potentially universal, biologically contingent, qualities of the active encounter.

Similarly, Salen and Zimmerman's four part definition of interaction separates a particular 'mode' of interactivity relevant to this project — that kind of interactivity emerging from designed rules (artificial relationships and restrictions) that gives actions within a system "meaning". This 'mode' is distinct from other 'modes' such as passive observation (the interactions of two colors in an image) and meta-contextual activities that surround game play (tournaments, trading, planning, etc.) (59).

It's noteworthy is that definitions offered by Salen and Zimmerman and others do not define, exactly, how interactive an artifact must be in order to qualify as "interactive". This flexibility is explicitly expressed by Chris Crawford in the *Art of Interactive Design* ~~where he states that even a door knob is interactive, if only a little~~. This variability is a necessity for Dominic Lopes when he posits that for a work of computer art, in order to best exemplify its kind, should ideally be "more" interactive (98). For this project, artifacts that are more interactive may described as exhibiting behavior in response to input that is more nuanced, subtle, surprising, demanding in attention, or, and more generally, promoting use that results in a more sophisticated and nuanced understanding of the artifact.

In contrast to these open ended views, Malcom McCullough's definition of interactive makes a distinction between items which are interactive and those that are merely operable (20). That there may be a common perceptual distinction between those things that are accepted as interactive and those that are not — even while the interactivity of such artifacts may be variable — has led this investigation to focus on simple artifacts straddling the line between the two categories in hopes that a perceptual distinction may become more apparent.

My experience is that direct encounters have the most immediate impact in cases where the artifact is designed to support direct manipulation. This kind of interaction is distinguished by a continual change in response to continual input. Simple (operable or manipulable) examples include scroll bars and moveable windows, complex examples are found in the control of an avatar in most action video games or the interface of a video editing suite. Direct manipulation is noteworthy in that it is often effective and enjoyable (Shneiderman). Steven Swink discusses this kind of continual manipulation as being central to the interactive "feel" in games (??). Despite it's aesthetic or utilitarian efficacy, this type of interactivity is particularly difficult to prototype via analog means (Snyder ??). Models that address this kind of interaction may be particularly useful.

The import of manipulation is found in Greg Costikyan's definition of games which has provided invaluable to my studies for a number of years. Among a variety of necessary elements he defines two of particular import; tokens and resources. Tokens being the elements of a game that are directly manipulated, resources those qualities that tokens are used to manage. In my experiences however, the distinct between the two is unstable and dependent on the perception of the player. I'll return to his later on page??.

Finally, Dav Svanæs's and Paul Dourish's writings on embodied interaction have been very informative. The ambiguities encountered with Costikyan's definitions were echoed in great detail in Svanæs's and Dourish's discussion of embodied interaction. This view, informed by the phenomenological philosophy of Marleau Ponty (and others) emphasizes the interrelationship of thinking, action, identity. Dourish (particularly his descriptions of coupling (138)) and Svanæs's empirical study on the movement of users' locus of attention (pp.) were particularly useful in providing an empirical and philosophical context to the unstable token/resource phenomena I had long observed.

From these writings and my experiences it appears that our attempts to locate the consequences of our actions — or ourselves even — in the midst of the artificial causal landscape of an interactive artifact may be a defining aesthetic characteristic of interaction. The sense of agency attributed to the computer is then a quality of the subjective experience opposed to an external prerequisite for interactivity. As such, instead of defining interaction in terms of "reciprocal action", I define single user, screen based, interaction generally as *encountering the consequences of one's own actions*. From this my model is based on the most simple of outwardly directed action — manipulation.

---

[1] Writings on game design have been particularly insightful and may be relied on more than would be expected.

[2] Many systems exist for building interactive artifacts from pre-existing components and/or pre-existing behaviors. However, these solutions often represent a limited realm of possibility and promote convention over innovation.

3 'Better looking' artifacts are often perceived to 'work better' (Norman, "Emotional Design" 17).

[4] Not a consensus view. Chris Crawford vehemently dismisses the use of narrative techniques as conceptual tools. (pp)

5 Toward the end of this project, I was introduced to an interesting study, the results of which I will discuss later: Interaction Gestalt and the Design of Aesthetic Interactions : Youn-kyung Lim, Erik Stolterman, Heekyoung Jung, Justin Donaldson

Move to conclusion?

intentionally destroying what Bolter and Gromla describe as the transparency

#
#
#: ***Interactivity***

Furthermore, while a user may view the artifact differently, this does not negate any value the idea may have for a designer. Similarly, in the visual arts a viewer need not have an understanding of **color theory** in order to view a work, but its utility to the designer is hardly diminished. In contrast, the view by a designer that an interactive artifact acts (let alone thinks) may actually have have negative consequences on their design decisions.

Disparity between a user's perception of an artifact's 'intelligence' alongside the artifact's actual capability to respond or act appropriately is a noted failing in a number of interactive projects. (Sharp et al.) This phenomena also has parallels in video game design where a dissonance may exist between the sophistication of a character's visual presentation and that of their behavior. The phenomena may be considered an impetus for "Max payne cheats only" by artistic duo JODI. In the work the digital character assets of the game "Max Payne" are digitally modified to produce a sometimes grotesque if not other-world views that echo the dissonant intersection of the human world with the foreign space of computation.

On a more abstract level this is the same problem with the design of interface metaphors. Oftentimes created to facilitate a user's intuitive grasp of an artifact's working, they can be counter-productive when the structure fails to match the perceived façade. Worse yet is when perceptually accessible part of the artifact is seemingly unrelated to its function. This disparity between the internal workings of systems and their interface led to the advocation of designing from the interface backwards, or more appropriately, from the user backwards. (Cooper ??, Norman, "The Invisible Computer" 23). This position rightly emphasizes the design of interactions in response to the user as whole person and to avoid forcing them to bend to the design. Without a clear relationship between design *goals* and executable *decisions* I wonder if this view unintentionally emphasizes solutions built on what users already understand — or worse, on what designers understand — opposed to potentially novel solutions that may work better or are at least more enjoyable. It may be that the design problem may not be best approached as "designing the interface first" but by designing the interface to communicate or express the system's function in a manner that rewards continued use by providing continually compelling interaction.

Manipulation

Definitions

The definition of manipulation is to handle or control, typically in a skillful manner. The root of the word being from the latin *manipulus* or ‘handful’ and *manus* hand. (It is also the root of the roman *Maniple,* a division of the army; one considered to be a ‘handful’). The term manipulation is also used metaphorically to refer to the control of particularly complex items, and even in a social context — the most ‘interactive’ of settings. Regardless of the specific use of the term ‘manipulation’ there’s reason to believe that our understanding of the phenomena is built on the less complex, body-centric, understanding of the term (Lakoff and Johnson). Lastly, as manipulations are predicated on causal relationships between items, these relationships themselves should have qualities that are in turn manipulable.

My definition of interaction then requires a strict definition of manipulation. For my purposes I’ll define manipulation as: The intentional bringing-into-alignment of some perceived quality of an entity to that of an intentional value. In other words, manipulation is an action with several criteria:

• There must be an actor with intent.
• There must be an observable, external *entity*.
• The entity must have some observable *quality* that may change.
• The observer’s locus of attention is on this change.

For example; turning my coffee cup so the handle faces me, the entity is the cup, the property its rotation, the intentional value is a preconceived rotation where its handle faces me, the manipulation as a whole the act of transforming its rotation to that of my ideal.

While I believe this definition useful, it is problematic in a number of ways that I will spend the remainder of this section addressing. Firstly there is no available method of quantifying ‘intent’. It will then be best to discuss the point at which an actor’s intent manifests (i.e. input). Next, and perhaps most problematic, is the requirement for a persistent entity with observable qualities. Not only are people highly capable of perceptual shifts that reframe fundamental perceptual starting points such as figure and ground[6], but the screen space may present visual forms with computational structures divorced from our common perceptual understanding of them; In other words, the entity-ness of screen based forms is highly dubious. Lastly, detailed psychological investigation of the mechanisms or manner in which a user’s locus of attention changes in response to action is beyond the scope of this project.

In the coffee cup example, the manipulation is composed of changes in any number of perceivable entities; an arm, fingers, the cup’s saucer; without being the locus of attention these do not count as manipulations. That our locus of attention may move between elements or actions of a larger task is a given. If our attention should move from one point in this causal chain to another it may be ambiguous whether this should be described as two sequential manipulations, two overlapping ones, or one manipulation with some allowance for the specifics of our attention. The distinction I'll leave open for now.

It should also be noted that our visual attention may be distinct from the locus of our agency, and that a visual element may draw out visual attention (through kinetic of visual means) while our locus of attention remains fixed (or vice versa).

In the sketch *Twins 01* two cursors respond directly to input. However, one cursor vibrates when the user gives no input. Once a user provides input, the behaviors of the cursors switch, the one that vibrates is calm, moving like a normal cursors would, while the other, previously still cursor, follows but with an overlay of wiggling movement. While the vibrating cursor may draw our attention, the feeling of 'under our control-ness' seems to belong to that entity exhibiting behavior most similar to our own; in this case the still, or continuously moving cursor.

Costikyan's definition of games makes a distinction between tokens (elements within a game we control directly) and resources (elements are qualities we manage via tokens). This is valuable conceptual tool, but more fascinating is that, in practice, what qualifies as one or the other is subjective and prone to change. This change where the perceived agency moves from one element to another I describe as a "Token Shift".

One common and peculiar type of shift is inward, where an action or manipulation is interrupted by an otherwise un-expected turn of events — or a misbehaving relationship in the chain of causality — forcing our attention to it. An older study of mine, *token switching* demonstrates this. In it a grid of cursors are variously 'activated' and 'deactivated' dependent on the (invisible) system cursor's location. An active cursor is tightly mapped to input, and a deactivated one moves to its original place in the grid. The objective result is that as the user moves a mouse different cursors will respond in the manner expected; the subjective and experiential result is generally one of unsettlement. (At the time I found it noteworthy that the experience of controlling the system is distinct or at least more acute than the experience of passively observing its use.)

This kind of shift in the 'proximity' of agency, even when the objective qualities are constant, is noted by Dav Svanæs. In his studies he observed a shift in users' perception of a simple interactive system — at least in the language used to describe the behavior of the system — their descriptions of the behavior changing from more independent (the computer acts) to dependent (I act) the more time spent interacting with the system. In other words, through use, the user's "loci of attention" moves through a continuum beginning with their body and ending with the changed element in the screen. (157)

Picture?

The study *TokenChain* is an example that allows users to instigate shifting. In this case though the shifting is 'outward'. The ability to manipulate the cursor at the center of attention is never removed, but instead the consequences of this manipulation are extended; the cursor's status as the focus of the common 'point and click' gesture is shifted through one element to another. As the chain extends, the focus of our attention moves with it. While the entity of the manipulation changes, the specifics of the relationships (input to output) do not. In contrast to the previous study, the experience of use is not unsettling, and in fact is barely noteworthy.

The two types of shifts I've described, 'inward' and 'outward' seem dependent on a perceived (if not actual) causal relationship where one element affects those further down the causal chain. In *token switching* attention is pulled towards an element that would otherwise be acted-with unconsciously. In this case it might be said that the change in the actual causal structure outpaces or disrupts our expectations, whereas in the *TokenChain* the reverse may be true; the change in the actual causal structure follow behind the change in our locus of attention and reinforces expectations.

That this 'shifting' is in some way an experiential phenomena with aesthetic dimensions — it is capable of prompting some kind of feeling — is clear; Its specifics less so. I am unsure, for example, as to whether to describe it as a single manipulation with a changing entity, or two separate manipulations that happen in sequence. An answer would be best based on a better understanding of both the manner in which our locus of attention changes, whether moving 'laterally' between unrelated elements in the same context, 'hierarchically' between causally related elements, or if such distinctions are even tenable. The specifics of this are of great interest to me but beyond the scope of this paper. It is enough here to note that at one point what was once a manipulation of one thing is now a manipulation of another, and that despite being contingent on subjective perception, these perceptual shifts can be instigated by the design of the system.

There are several known visual cues that we use to distinguish one thing from another including proximity, connectedness, and sympathetic movement. Things that are close together often go together — or at least affect each other. Things that move together are often seen as part of a surface — typically on a solid object. Kinetic cues and otherwise 'static' visual cues may easily disagree; for example, in an instance where two distinctly separate dots move in unison — imagine a dark night watching a car's headlights from afar. While this conflict in cues might be cause for a little tension, it would hardly be described as off putting or uncomfortable. However, when the simultaneous movement of two, visually distinct, dots are under the control of a user, it's possible — in situations where the distinctness of the dots would imply a difference in behavior — for any subtle tension to apparently increase. In *twins_02ab*, either a bar or two 'connected' dots may be dragged within the confines of the space. When the bar's movement is limited by the constraints it seems perceptually neutral, unremarkable, perhaps natural. Whereas when the two dots are stopped by their constraints there seems to be at least a momentary tension, as if we expect the two elements to behave independently. A following study, *twins_02a* similarly contains two circles that will move together when clicked and dragged, however each is confined to half of the space. Once one dot collides with its bounds the dot's movement will become restricted whereas the other may continue to move freely. Once one or the other exhibits some independent qualities, there seems to be a release in tension. It would seem that while our mind may give the kinetic and manipulable qualities of a form or forms some precedence in regards to its status as an 'entity', it does so perhaps grudgingly or with reservations.

Another study is built with similar behavior. However the un-clicked circle has noise introduced into the mapping between the mouse and its position resulting in a wiggling when dragging. At various times, its direction of movement will be either complementary, opposed, or tangental to the direction of input movement and also to varying degrees. The result appears as an ambiguous causal relationship where the tightly mapped dot affects the wiggly one. However, at times it appears as if the dragged dot should be responding to the wiggly one, creating slight tension when the dragged dot doesn't conform to expectations established by the visual relationship. This is somewhat more prominent in *twins_03* where a line is shown connecting the two while dragging.

In *twins_04* a collection of dots is used. When a dot is clicked, other dots will move somewhat sympathetically but with limitations. The initially clicked dot is tightly mapped to input and responds immediately and exactly. Another selection of dots will move as if tightly mapped, except they will move in fixed increments of distance (that of their width), and only if the velocity of the input movement is above a certain thresh hold. They have the appearance, or feel, of being 'sticky'. Another selection of dots moves more continuously, but lags behind mouse input movements. While the clicked dot always responds immediately and continuously, the focus seems drawn to the collection of 'sticky' dots and their somewhat haphazard response. I might describe the overall effect as a 'tease' where we anticipate (or perhaps attempt?) a repeatedly deferred token shift.

It may be reasonable to say that in a more complex artifact with many things responding to input, we look for or notice what is at the tip of our conscious control, and that the target of a manipulation may be influenced or redirected by the manner in which the system responds. In the cases like *TokenChain* where a user may consciously switch their objective point of manipulation from one element to another it may be said that a form's manipulable qualities may themselves be the target of a manipulation.

plasticity

In the screen space there is nothing a priori manipulable. As screen visuals and their behavior are materially the result of logical algorithms, some have argued that this constitutes the essential reality of the artifact and that anything beyond that is 'window dressing' (Koster). In reality, the specifics of the inner workings of these creations may be unobservable and have no baring on how we subjectively understand them or attempt to interact with them. The math and logic used to display a black square on the screen are secondary to our experience of it as such, unless its computational description have direct baring on what qualities of it are manipulable.

For there to be manipulable form, the form — and the manner in which it can transform — must be created. This is typically done through the combination of mathematical expressions and programming logic that defines, in the end, a visual form commonly referred to as a parametric form (McCullough 168, Reas et al 93). Such abstract forms have quantifiable parameters that define a range of potential observable forms they may take on. The continuum of forms may be referred to as a possibility space.

figure?

It is entirely possible for two parametric forms to share one or more specific overlapping expressions. At these points a visual form's manipulable qualities may be ambiguous. It may also be possible to switch between two different parametric forms. The quality that was once manipulable disappearing in place of another quality (or even an entirely different entity) without any visual discontinuity.

In the study *interactive figure ground* a rectangle is divided into half white and half black. It may also be described as a white square on a black background and vice versa. By moving the mouse the system changes from one structure to another, switching between them at the point of overlap (where the rectangle is exactly half black and half white). The study *Fidelity* is another, slightly more complex, example.

Figure x: interactive figure ground

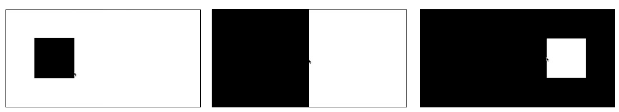
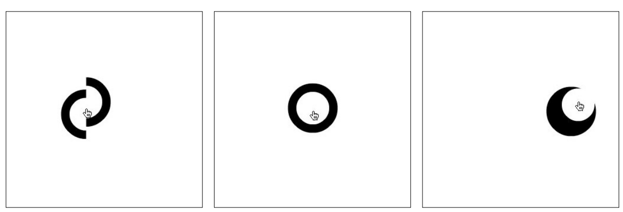


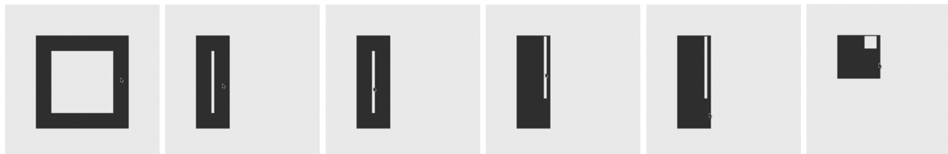
Figure x: Fidelity



It's also possible for the possibility space of two or more parametric descriptions to overlap at more than one point; even overlap completely. *Ring Box* allows a user to manipulate a form in one of two possibility spaces; one where the individual 'bars' of the form can be moved via a click and drag, and one where the negative space in the middle may be moved as if it were a solid form. This phenomena is found in almost any piece of software from word processors — allowing for the manipulation of their content as-language (in the case of typing) or as-image in the case of setting type face, type weight, margins and other formatting variables — to 3D modeling programs that support the manipulation of form in terms of points, lines, or surface.

Bigger!

Figure x: ring box



In these examples there is a clear continuity of controllable transformation, but if there's a specific entity being manipulated it is somewhat ambiguous. The most general description would be that these are simply parametric forms with unique manipulable qualities. However, these situations where the same or similar action has different consequences could also be referred to as modes (Raskin 37). Furthermore they might also be described as two forms that support instantaneous "transcoding" (Manovich, "Language of New Media", 45) in that the perceived form is bounced between two different mathematical or numerical models (though its possible for a form's mathematical underpinning to remain constant even while a person experiences a perceptual shift). In my definition each study might be described as supporting a 'token switch' between two manipulable forms that have momentary visual similarity, or a switch between tokens coinciding with an overall change in the elements of the artifact.

section / page

While we may have a single locus of a attention, it may be possible to manipulate multiple things at once so long as we can make some attempt to abstract them into a 'unit' (this question I'll return to later).It may be possible that, given an increase in quantity of complexity, the manipulation of a collection of things (even visually continuous things) might become the manipulation of a gestalt level quality of a larger 'whole'. The above example may be such a phenomena.

A detailed analysis of digital form — being that form predicates interactivity — would seem to be a necessity for a proper description of interactivity; unfortunately the variety of parametric form is limited only by the creator's faculty with math, logic, programming and available computational power. While it may be logical, or colloquial, to in turn give something like "the pixel" material status, the pragmatic view of computational aesthetics emphasizes the algorithms and programming fundamentals — assignment, conditionals, loops, and functions — that change these pixels (Reas et al. 13). Because of this it would be ideal for a definition of screen based interactivity to bridge the language of computation and a perceptual based language of interaction.

That the manipulable qualities of form share some perceptual similarities regardless of the specific form is implicit in the phenomena being named at all. It should then be possible to describe perceptual or formal changes between forms stemming from a users's conscious action with a common language regardless of the specifics of the form. That we are mentally capable of understanding novel forms with novel manipulable section / page

qualities (learning software or new game mechanics) implies an ability for abstracted reasoning about causal relationships. As such, it should be possible to outline various general qualities particular to manipulations that are applicable in the design of manipulable form regardless of the forms' specifics.

For a model of manipulation to be useful, these qualities should be subject to expression and modification in formalized terms. A designer should not only be capable of describing an artifact's manipulable qualities, but also of 'manipulating' these qualities in order to produce a better artifact. The ability for a manipulation to serve as the target of another manipulation may also be embedded into an artifact itself allowing for more complex artifacts with emergent gestalts.[7] Together this capacity can be described as meta-manipulation.

Meta-Manipulation

Theoretical implications and difficulties of using a manipulation as the target of another manipulation has implications for the work in the "studies" section.

The most important aspect of a manipulation as defined so far is the requirement for an entity — specifically a quality of an entity — to be acted upon, and the subjective contingencies of its entity-ness. The ability to manipulate the target of a manipulation can thus be accomplished by allowing a user's perceptual 'target' of a manipulation to ebb and flow in response to the dynamic gestalt. As such, the studies generally utilize a collection of many elements that all respond to input.

Concerning quantifiable qualities, I initially focused on what I felt were two intrinsic and readily apparent aspects; that manipulations are generally not instantaneous, and that there typically exists a quantifiable relationship between the observed change and the change at the point of input.

By my definition a manipulation has a distinct beginning (the formulation of the ideal state) and end (the resolution of the transformation) — the time between lends the manipulation a temporal dimension. The time between the formulation of the ideal state and the end of the manipulation I describe as latency. Latency is typically defined as the time in a system between input and any response at all. This is commonly studied aspect of human computer interaction as variation in latency is easily observable when present, and when above a certain threshold it will diminish, if not destroy, the perception of interaction (Swink 45). By the strict definition there is little room for a decision to be made about latency; the lowest latency possible is generally preferable. In my definition however, it may be used to describe manipulations that take more or less time to resolve. I'll refine my description of a manipulations temporal qualities later, but for now I'll use latency to refer to the time between first input and the resolution of the manipulation.

The second quality stems from the temporal component. As a manipulation takes time the manipulated quality should then change over time between it's beginning and end state.

Furthermore, this change may not have a one to one correlation to the input. It need not even be a linear relationship. For example, a scaler value such as a cursor's horizontal position may drive a binary value like 'visible' via a mathematical expression such as (mouse horizontal position > 100) where the result is equal to 1 if the position is greater than 100 and 0 if otherwise.

In order to discover further ambiguities inherit in the model or unexpected consequences of its application I created a series of studies (described in the following section). In applying the model I was forced to articulate manipulable qualities in a manner that was computationally meaningful. Alongside with the studies I began formalizing the model both as a shorthand notation and in the form of a code library that would support the articulation of relationships with increasing brevity and flexibility.

---

[6] See Wittgenstein's duck-rabbit image or other classic optical illusions such as the old lady / young lady image.

[7] This quality is essential if the model is to be useful in supporting the kind of emergent complexity typified by definitions such as "meaningful play" by Salen and Zimmerman. As a valuative term, Meaingful Play refers to how choices in a game system have discernable consequences. I might word it more specifically to say it refers to choices that have discernible affects on *future choices*.

Studies

things

My initial explorations looked at offsetting the entire duration of a ‘manipulation’. For example, with zero offset, a cursor would would move in tandem with mouse input, with a larger offset, the cursor would complete the same movement(s) in the same amount of time and in the same manner, but would begin at a later point in time. Furthermore I would create numerous cursors, each with increasing offset. (Latency studies 01, 02.)

In an attempt to touch on mapping I then introduced noise into the relationships between input and position. In doing this I accidentally introduced variation into what I came to call the manipulation’s *tolerance*. As the ending of a manipulation would be qualified subjectively, it follows that some results may be ‘close enough’; that in turning my coffee cup to face me, it need not be rotated *exactly* ninety point one degrees, so long as its new orientation is functional in the context of the initial manipulation. In sketch latency_03, there is progressively more noise introduced into the both the value space the position values move through, but also the values that they end at. In latency_03a, the latency from the first studies is removed leaving only the noise in the value space and tolerance.

In latency_04 elements with less latency have more noise in their mapping, and elements with more latency have less noise. The results are peculiar in that the elements are both ill responsive (perhaps frustratingly so if it was in the context of some utilitarian application) but are comforting (for lack of a better word) in distinct ways. Tension is created between immediate but ambiguous response vs. clear, but delayed, understanding. I’m not sure if either behavior could be said to better ‘echo’ the intent or input of the user.

05 swaps the mapping between x and y of half of the cursors which otherwise have increasing latency. The cursor with the most direct mapping is easily — and comfortingly — found, but the cursor with the next lowest latency draws attention to itself despite the inverted positional mapping. I imagine we rely most on traditionally defined latency to discriminate between the effects of their actions and otherwise independent events. In other words, causal proximity might be more or most important for establishing our locus of attention than visual or kinetic similarity. This could be an area of future investigation.

Until now, elements had consistently increasing latency. In 06 the difference in latency between the closest mapped element is relatively high, but the difference in latency between it and the next decreases steadily. The experience of use is dominated by its kinetic qualities, a flurry of action that follows the initial exploratory gestures.

07 is the same as 06, except a property, rotation, is mapped to the ‘direction of movement’. Instead of mapping one value directly to another, a property (rotation) is mapped to something more formulaic:

the angle between its current position and another position. The distinction between a ‘measurement’ or more ‘absolute value’ such as position and a more formulaic or multi-variate property like ‘angle-to’ is, in reality, arbitrary (as a screen based form, it may be described as entirely formulaic). The distinction is contingent only on the data structures in the programming environment, which may not manifest in clearly observable ways. So there’s no real reason to create such a distinction. Here though the distinction between the existent computational qualities and the mental faculty of the artist/designer comes into play. It may be ideal to make manipulable a quality of an element that is clearly perceptual, but has no computational equivalent. For example, the distance between two corners of a box may be relevant to a designer’s idea, but have no analogy in the code base at hand. Inversely, the computational environment may support properties that are otherwise unapparent to a person, either because of the level of their programming skill, or simply their creative and subjective way of perceiving the screen space. (This issue comes up again when working on the application.)

move to Formal Expression : time?

~~Secondly, the manner in which delay has been introduced programmatically means that any property has not only a current value, but a value at any given previous time. In a sense the manipulations of the delayed cursors aren’t delayed so much as they are being driven by a property-as-it-was, i.e. the mouse-position-*n*-steps-ago.~~

In the end, the cursors’ rotation gives them an entirely determined, but perceptually independent quality.

At this point I re-factored the code for the initial sketches to make future sketching/coding easier (latency_08). After these changes the system also supported a number of new qualities. This was the beginning of the process of formalizing (and implementing) the proposed model in order to provide computationally analogous statements about manipulation. This is discussed more in *Formalized Expression*.

section link

! broken?

Latency_09 makes use of varied frame rate — essentially affecting latency in the traditional sense. Holding down the mouse button decreases the frame rate, releasing increases the frame rate back to normal. While the tightly mapped cursor was originally easy to spot, as the frame rate for the collection drops uniformly, the cursor becomes increasingly hard to identify with.

In 09a, holding down the mouse button decreases frame rate — or increases traditional latency — non-uniformly. Cursors with more delay have higher frame rates making them move smoothly, while those with less delay have lower frame rates; their movement stuttering. Once the mouse is pressed and held, to me it appears that the smoother moving cursors draw attention to themselves despite being clearly separate from input. *latency 09b* and *latency 09c* are variations on the way stuttering is staggered.

latency study 10 provides for delayed manipulation of one set of an elements qualities (position), but more immediate manipulation of another — rotation. When the mouse button is pressed, the rotation of each cursor simultaneously animates 180°. Triggering the rotation gives the artifact as a whole a certain unity — while otherwise appearing more as a 'collection of elements'. As a corollary to the adage of visual perception "things that are close together, go together", it might be said that "things that act together, go together."

This idea I returned to later in the short series named 'finger studies'. In these studies — built from the initial latency studies — an "invisible button" sits at the center of the screen. In the first study cursors that touch the button change to the familiar "finger" cursor. In the third however, all the cursors change when the real cursor touches the button. Here the cursors are disrupted by the increased delay, but re-asserted in some fashion once a different parameter visual becomes immediately manipulable.

latency study 11 is similar in concept; the rotation and position of each cursor is directly modified by mouse input with the modification of their positions being delayed. The rotation however is driven by the angle to the most tightly mapped cursor position. Here the delay on the cursors' position results in variation in their rotation. In contrast to the previous study, the immediate control over their rotation makes the cursors appear — at least during initial tinkering — to be *more* independent. This rotation though is also dependent on their positions which, while also driven by user input, are so delayed as to appear independent. In this situation where a change emerges from the combination of perceptually independent activity (the movement) and a dependent variable (the position of the tightly mapped cursor) the result seems lean toward causal separation.

The twelfth study was an attempt to have cursors driven by mouse movement in both a delayed fashion and by immediate movements. In other words, the position of each cursor is driven by both the cursor's current position relative to the real cursor and the previous position of the real cursor. The results were ambiguous and had unintentional glitches. Many of the cursors with higher delays rapidly flicker between two positions. Having a single value driven by multiple creates complications I'll discuss later.

section link.

latency 13 allows the mappings' latency to be modified by mouse press. Pressing and holding continually decreases the latency resulting in cursors accelerating towards the mouse until they are all moving together. Releasing the mouse button returns the delays to their initial, staggered, values. Lower levels of delay result in mouse-trails.

The objective at this point has been to isolate and explore interactive qualities to the exclusion of visual variables as much as possible. (Avoiding variation in the studies' kinetic qualities while investigating direct manipulation would be much more difficult if not impossible). The next few studies (14,15,15a,15b, and 15c) look at how any of the relationships in these interactive systems might be retained while modifying or replacing the visual forms related to them. It may be interesting or helpful — as exercises for a screen designer — to switch back and forth between isolated 'visual' properties and the system's 'manipulable' properties in order to promote both flexibility in thinking and better understanding of the relationships between an artifact's computational and perceptual qualities.

user driven meta-manipulation

As mentioned, the qualities of a manipulable form should themselves be manipulable in order to support design decisions. However, it should also be possible to embed such meta-manipulability in the artifact so that users can themselves modify an object's manipulability. Such reflexive relationships are simple to create, but very unintuitive to describe in terms of meta-manipulation

drag box inverted

The most trivial example is a draggable box. Here a relationship exists between the cursor and the box such that the box moves when the mouse moves. However, a another relationship exists between the box's pressed-ness and the active-ness of the position relationship. The result is basic drag-and-drop, the cornerstone of graphical user interface design and experientially banal. The relationship meta-manipulation is more apparent if the mapping is inverted so that clicking the box deactivates the position relationship.

Meta-mapping may be more circular however, at which point subtle ambiguities appear.

link

In *drag box 01* the cursor affects the box position, and the box position affects this relationship. The box may be moved to a specified point to disable to the ability to move it. drag box 01a is similar, but here the box's position may be moved, after the fact, away from the position that results in disabling. The experience in the former, where the relationship between cursor and box is broken permanently seems to engender a slightly stronger sense of agency than the latter where the experience is one of constraint.

these kinds of relationships require conditionals.

The formal expression of the relationships seem to apply to both behaviors. However, it may be more accurate to say the position of the cursor drives both the box and the activeness of that relationship. The cursor then may always move so that the position relationship is restored, and the box returns to its following behavior.

In the first example the mapping is dependent on the input. In modifying a manipulation it should be possible to also change the property being manipulated. In the simple study *box 02* input changes the boxes's target manipulable quality from position or rotation.

As it should be possible to manipulate the characteristics of manipulation's mapping I created another study where the default 1:1 mapping is changed to 2:1 (the output is half the input) when the box's position reaches a specific threshold. The result is a feeling of the box encounters resistance when dragged the wrong way past a point, like rubbing soap against the grain of a shark's skin. It also had the feeling of pulling something through a membrane. I was compelled to adjust the visuals to abstractly represent de-boning a chunk of meat. One de-boned, the left over form can be dragged with impunity.

As the resulting interaction is at least initially surprising, it's difficult to say that the qualities of an ongoing manipulation are intentionally being "manipulated". However, it also seems entirely intuitive to state — once the relationship is apparent — that the box's "draggability" is a function of its dragging, and that a user who desires to create a 2:1 mapping may do so very intentionally simply by dragging the box to the marked threshold.

I suspect that a talk-aloud experiment would show that users describe the behavior in terms of the computer, or an element in the system, affecting the interaction, opposed to the user stating that they were "slowing down" the box. My question would then be if it is possible, and under what circumstances, for a users' "body space", to use Svanæs term, would expand to include the notion of "manipulating" the qualities of an ongoing manipulation. Such a conception might be possible but require substantial use time constituting the markings of "expert level" knowledge. Furthermore there is research that suggests that people view action and reaction in a relatively discreet number of combinations of an actor performing an action (Pinker 219). It may be that the appearance of resistance is always considered as the result of "two parties" even if one is inanimate; in such a situation I'd be very curious about how the idea of a locus of attention resolves.

Describing these behaviors in terms of meta-manipulation may not be immediately intuitive. But it may be more so than translating the behaviors into traditional programming logic.

Finally, another distinction that arose in my mind from this study (and from the rotation of the cursors in *latency study 11*) is between a manipulation that requires constant input to bring it to completion —like dragging — and one that may be have a duration but requires only a single, relatively instantaneous, action — like clicking. It seems that two manipulations may require variations in potential input while the the range of values that the manipulated quality will pass through could be identical. As such it'd be worth distinguishing between the value space of a manipulation (the range of values the quality of an entity may go through) and the effort space (the range of input or activity required to achieve the result. Traditionally, changes that happen due to user input, but that are not guided by input can be referred to as triggered animation. This will be discussed in the section *Formal Expression : Time*.

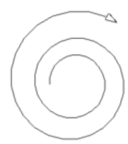
drawings

Drawing to the screen — albeit repeatedly at superhuman speeds – is the root of screen-based, computer generated, form. Couching computation in terms of the 'manipulation' of 'drawing tools' is an approach to making programming more approachable, and is similar, if not the same as, the drawing done in LOGO via the command controlled movement of the 'turtle'. For the purposes of the model presented in this thesis it may provide a way to partially accommodate the plasticity of the screen space.

Manipulation of form vs the (continuous) creation of form.

Source wikipedia

Figure x: LOGO's turtle leaving behind a line as it travels.

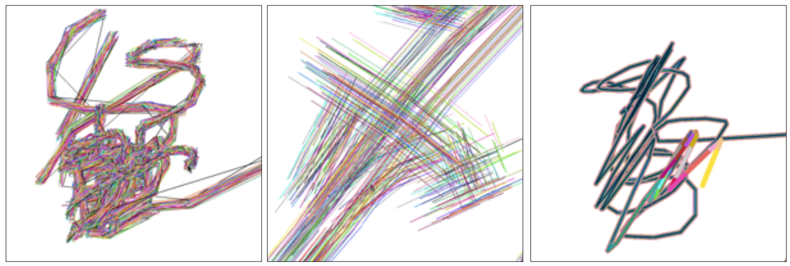


The proposed model contains language that applies to both computational and perceptual contexts — how its use results in emergent qualities may be still be distinct in the two realms. For example, as stated earlier, it should be possible for the manipulable qualities of form to be subject to manipulation — and that current manipulations may affect later manipulations. In computational terms this takes the form of a quantifiable changes to the described qualities of a manipulation. However, in the perceptual realm, the qualities of a manipulation may be contingent on previous ones by virtue of a change in context. The act of drawing might involve no change in the manipulable qualities of an entity (a mark maker) while the use of its manipulation may be contingent on previous manipulations (previous marks). Thus manipulation involved in making the first mark on a page is objective the same as making the last mark, but the perception of the two actions may be distinct due to the visual context of the mark making. The characteristic of the manipulation changes for subjective reasons.[8]

This is similar to the emergent interactivity I discussed before; where continual interaction and familiarity change the subjective understanding of gestures that have objectively remained unchanged. In the latency studies, different movements often created different kinetic qualities, in the case of drawing, different manipulations create different visual qualities. In another similarity to the latency studies, the following drawing studies allow for the manipulation of a number of mark making entities. They remind me of the childhood activity of drawing with a fist full of crayons.

In contrast to this physical activity where mark makers are manipulated 'as one' the digital equivalent allows for a great deal of variety in the way input is mapped to their changes. The results, even more so than the as with the latency studies, is a general perception of manipulating "stuff". Perceptually, it's sensical to speak colloquially about the manipulation of a range of multiples — singular entity (a hair), collections (hairs), stuff (hair). Computationally, this common desire is addressed through loops where the a set of operations on a single piece of some is serialized; the computer's ability to execute the instructions quickly and repeatedly resulting in the perception of something being done all at once and continuously. This approach is useful and powerful for many reasons I will not go into, but is also very distinct from the perceptual results.
For these reasons, this section looks more closely at the complexities of quantity in the context of the manipulation targets ~~— particularly in the computational context.~~

*Figure x: **drawing 01**, **drawing 02**, and **drawing 03***



At this point the implementation of the proposed model (the code library) kept constant track of the change in any value it was watching, effectively granting access to the velocity of things like the mouse position. While I was trying to avoid changing manipulable properties in response to previous actions, I did start utilizing these time based variables to change the pens' mappings.

That an entity may have a quality is conceptually reasonable. That a quality may itself have a quality (such as 'rate of change') is a little less obvious. However, such real life statements such as "The wheel spun faster" or other changes to an entity's kinetic qualities are perfectly intelligible. It'd also be understandable (if not wordy) to state "mouse position changes wheel rotation speed". Such a meta-quality 'velocity' may also serve as an entity as in the *meta*-meta-quality 'acceleration': the rate of change of the *rate of change* of a value. It may be desirable to amend the proposed model so that "a quality of an entity may itself serve as an entity, in which case it has the quality 'rate of change', which in turn may also serve as an entity (ad infinitum)." While entity-ness is subject to framing and perceptual shifts, it'd be hard to argue that a perceived visual entity is experientially similar to the perception of 'speed' as an entity. Perceptually, it seems more direct to refer to velocity and acceleration as qualities of the initial entity. However, developing a more discerning view is foundational to traditional animation studies. In the seminal *Illusion of Life* Ollie Johnston and Frank Thomas posit that kinetic phenomena such as "slow-in and slow-out" (movement utilizing acceleration and de-acceleration) constitute some of the core principles of animation. So, while the expression of kinetic phenomena in meta-quality terms may not be entirely descriptive of a lay person's perception, developing such discerning distinctions may be valuable for the practicing designer.

In drawing 02, the pen's is-drawing-ness is dependent on the velocity of a cursor; fast movements make lines, slow ones do not. There is no 'is-fast' meta quality to velocity though. 'Fast' is relative. To express this, the mapping quality would need modified so that scaler values are converted to binary values.

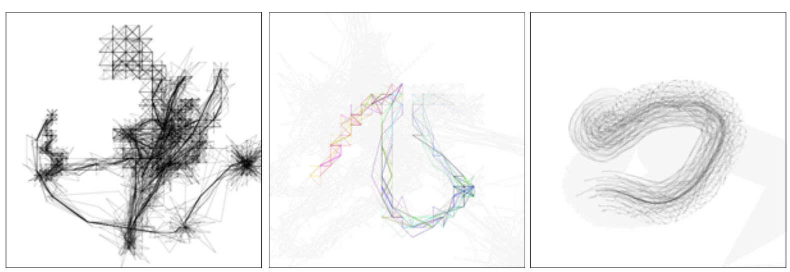
In drawing 04, the velocity of the cursor drives the amount of noise in the mapping of the pens' position; moving the cursor quickly results in lines being draw erratically.

In these cases, making the mapping of a manipulation itself manipulable presented a powerful technique for creating variation and more elaborate kinetic qualities, but revealed some issues with the concept of mapping in general. Specifically, if a mapping itself has qualities, what qualities does it have. In drawing 04 it'd be most accurate to say that the cursor velocity was driving a 'noise' quality of the mapping. To say that the mapping of any relationship has a "noise" property is problematic however. In fact, to say that all mappings have a common set of properties is impossible. What quality could be transformed so that a scalar value is converted to binary like in drawing 02? Each seems to require mathematical notation. This will be discussed more in the section *Formal Expression : Mapping*

The experiential result of these studies is more similar to manipulating a novel, reactive, form than to the experience of drawing. If the older, previous marks, were removed in some fashion, and the user left with just the most recent marks, the results would appear even less as a drawing, and more as 'reactive' or computationally generated form.

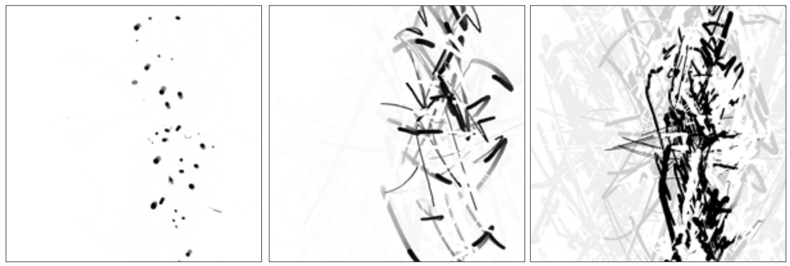
Computationally generative form, even when it appears static, involves repeatedly drawing and clearing an image space. Combing this erasing — even if behind the scenes — with the delayed and alternatively mapped pen manipulation creates a variety of manipulable form. So far I have only considered the manipulation of explicit entities, but it may be reasonable to look for universal situational properties, or generalized properties of an artifact, such as 'mark permanence' that can be the target of a manipulation in order to allow for these kinds of artifacts. The following studies explore this and similar situations.

*Figure x: **drawing 05a**, **drawing 06**, and **drawing 07***



The final two studies allow the amount of 'erasing' to be controlled by the user. The more compelling one is presented below.

*Figure x: Samples from **drawing 10***



In addition to issues in mapping, the drawing studies helped foreground a perceptual distinction between the manipulation of a thing, the manipulation of things, and the manipulation of 'stuff'. Along with this, a personal desire to more accurately describe relationships between one-and-many, and many-and-many.

In the early Latency studies it became quickly apparent that while it was possible to focus on a directly manipulable entity in crowd of similarly behaving items, focus often seemed drawn to the artifact as a whole. Here with the drawing studies In the absence of any controlled, visual, entities this phenomena is more pronounced. Perceptually, it appears as the manipulation of 'stuff'. Presently, my ideas focus on the manipulation of a 'thing', or perhaps the manipulation of multiple 'things' where the attention moves rapidly from one to another. The use of a collection of things however gives the artifact as a whole a unified gestalt, but also the sense of manipulating something amorphous and indistinct. Research in linguistics has found that individuals make a particular distinction between "collections" of items and amorphous "stuff" (Pinker, 167–174). The perceptual shift from the manipulation of a thing, to things, to stuff is likely not continuous, and could be the focus of further exploration. Regardless, language of manipulation (and its technical implementation) should support the creation of mappings between many things.

---

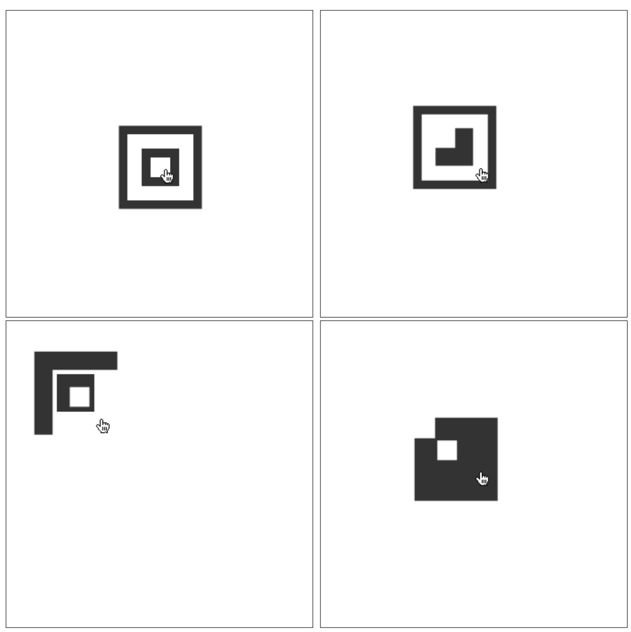
[8] It is also very easy to make an imaginative leap to ideas where previous marks have very real effects on the ability to manipulate the pen. Games like 'Snake' and 'Tron' come to mind.

Application

One potential benefit to a quantifiable model of interaction is the potential to represent and manipulate interactive relationships. A full application for the creation and modification of interactive qualities would be an undertaking far out of scope for this project. However, considering how meta-manipulation might manifest in a user interface helped reveal intuitively desirable functionality and problematic issues in formal expression. While the attempt entailed a number of general interface design problems, this section will focus on those issues most relevant to the investigation of the proposed model of interaction.

Like the studies, the forms utilized in the prototype would remain simple. While the model is agnostic to the forms being manipulated, I found that restricting entities to primitive black and white shapes useful. Overlapping them allows for easy construction of the kinds of novelly transformable visual forms created by computationally based parametric structures (see relationships 05.) In the event that the model is used in a classroom setting, such constraints might be effective in limiting investigation to existing forms (opposed to generative form), while promoting novel visual transformations.

Figure x: relationships 05



(figure needed)

page

section link

section link

Once I began working with these boxes, an initial desire was to fix the spacing between two boxes while letting them remain draggable.

(example)

While this is possible, it required several relationships to be created between the boxes' position and scale properties. Hardly elegant. Ideally the space between should be selectable somehow. (The existing equivalent is the gap tool in Adobe InDesign.) An old ideal of interface design is to "make it visible" (Norman ??), to present to the user the actions they can undertake and the objects they can manipulate. It may be more useful to invert the rule: Users should be able to manipulate what they perceive. As we can readily look at the relationships between entities as entities themselves; it stands that a user should be able to select things such as 'the angle between items', 'the distance between items' or even 'the relationship between corners'. My solution for the immediate problem was to create meta properties like top, bottom, left, and right. A more general solution would require a generic computational construct that could represent a variety of such relationships and a method of showing them intelligibly. (See *Formal Expression: Sets*.)

Another particular interface problem was the cursor. While it may be easy to form connections between different elements, any "interaction" is predicated on some kind of implicit relationship between the artifact's forms and the input. A designer would then require some self-reflexive ability to point to the point-of-input — the input which was being used to manipulate the tool in the first place. While an interface that is completely based on direct-manipulation sans hot keys or other accoutrements would be ideal, this introspective need to "refer to oneself" using the artifact at hand interferes with the potential for a real-time all-the-time editor. There will inevitably arise a moment when a user will need to either suspend the artifact, or step outside of it. Having two distinct modes is unavoidable.

In other tools this is handled by entering and exiting 'editing' mode or 'locking' the tool so that changes may not be made. The design challenge is then to both minimize the interruption so that a continuous work flow can be maintained yet make the distinction between the two modes clear. (I attempted to minimize the problem here through the use of a 'quasi-mode' where editing requires holding the 'e' key.) Ideally the tool's cursor could become detached from some kind of meta-cursor when entering the edit mode. This requires decoupling the default cursor from mouse or trackpad input, a feature otherwise known as "mouse lock", which is currently unavailable at the time of writing in the development environment I began in (Flash).

Once I was able to select and create a relationship from one property to another I immediately desired a method of selecting several at a time and create relationships between them, effectively grouping them. Expressing this computationally was initially problematic as the result was a series of circular references and an ambiguity about how the cursor position (when dragging) would be incorporated. This situation is detailed more in the section *Formal Expression: Mapping*.

The problem also exists in animation software like Cinema 4D where users can take advantage of both key-framed animation techniques and also physics based animation. In these systems the priority and even weighting given to various inputs can be set explicitly by the user. This would be useful functionality in managing internal relationships between elements that overlap with relationships with external input values. The solution I pursued was to give changes from outside of the system priority, letting them override the effects of other, internal relationships.

make new annotated video recording

(wip Prototype: Video**)**

Formal Expression

In the previous studies, statements about manipulability often had to be translated into the typical computational language of variables, loops, and functions. However, statements made about manipulation should be applicable both perceptually and computationally. A statement such as “the mouse position changes the box position” should have clear computational equivalent with as little translation as possible. This means formalizing perceptual statements about manipulation. This process alone raises new and unexpected ambiguities in the model that may not have been addressed in the previous sections.

relative vs. absolute

The simplest formal statement about manipulation might be the one mentioned above.

```
mouse horizontal-position -> box horizontal-position
```

Spoken thus: “mouse horizontal-position *drives* box horizontal-position”. In these expressions, the left hand entity/property pair is referred to as the “driver”, the right entity/property pair the “driven”.[9] The statement as a whole I’ll refer to as a “relationship”. In the context of the proposed model, a relationship is the the most basic formal description of an entity’s manipulability — the quality that allows for a manipulation to occur.

Even the simple relationship above may be interpreted in at least two ways. Interpreted as an “absolute” relationship, the value of the box position would be a exactly equivalent to the mouse position. Interpreted as a “relative” relationship, a change in the mouse position would be reflected by a corresponding change in the box position.

The previous studies almost always relied on such relative relationships. I find it more intuitive when such an expression results in the behavior “it moves as I move” than the behavior “it sticks to me”. This interpretation also allows for the target of the manipulation (box position) to change to something else (the position of a circle) without the unintended effect of the circle ‘snapping’ to the current mouse position.

example

see dragging examples

While the relative interpretation seems preferable, each of these interpretations may be useful. (I’ve found that absolute relationships are desirable when binary properties such as mouse pressed-ness drive things such as a relationship’s activeness-ness.)

example

Each interpretation also has distinct if not tedious computational aspects with more subtle behavioral implications. The absolute interpretation would be similar to a pointer, a computational construct whereby the value of one variable can be set to be that contained in the memory space of another variable.[10] In other words, by setting one variable equal to another, the value in the later will always be that of the former. Under this logic, the value of for b would be 2 at the end of this example:

```
a = 1
b = a
a = 2
```

In contrast the 'relative' interpretation would be expressed as such (where b would have a final value of 1.)

```
a = 1
b = Δa
a = 2
```

This logic is problematic as it does not have a direct computational equivalent in the languages I am familiar with.[11] It would require either the value of b to be set via calculation each time there was an assignment to a (which would override the assignment to Δa), or the previous value of a would need to be stored implicitly each time the value of a was set so that use of the variable b would provide the result of the difference between a's current and previous values.

Because all the languages I'm familiar with are imperative.

From what I've gathered, the functional programming paradigm typified by lambda calculus may be very relevant here.

A manipulation's latency property complicates things further. As the amount of latency may change, a continuous record of previous values for the driving property will need to exist. A computer program that functions this way would need more storage space the longer the artifact runs, potentially running out of memory. Alternatively, there would need to be a limit to the record of values, thus limiting the value of a relationships's latency to what the resources of the hardware can accommodate.

As it may be ideal to differentiate these interpretations, relative relationships will be notated in terms of the properties' deltas:

Δ mouse horizontal-position **->** Δ box horizontal-position

feedback loops

While relationships may be most intuitive when a user input value as a driver, this is not a requirement — relationships may exist between properties of a artifact's entities. This allows for the kinds of token chains described previously. Such a set of relationships may be expressed with multiple statements such as:

Δ mouse horizontal-position **->** Δ box [1] horizontal-position
Δ box [1] horizontal-position **->** box [2] horizontal-position

However, multiple statements allow for ambiguous expressions such as:

Δ mouse horizontal-position **->** Δ box horizontal-position
Δ mouse vertical-position **->** Δ box horizontal-position

The above could be interpreted so that one of the statements negates the other, or, that both statements stand with the the resulting changes in the box's horizontal position being the average, sum, or some other function of the two values. In this case there is still ambiguity. The first interpretation is likely the most intuitive, but the latter may also be desirable. The situation is even more problematic when a series of statements become circular. For example:

...
Δ box [1] horizontal-position -> Δ box [2] horizontal-position
...
Δ box [2] horizontal-position -> Δ box [1] horizontal-position

In this case, a change to some the first box's position affects the position of box 2 which in turn affects box 1, resulting in an ambiguous feedback loop. I've personally encountered similar problems before in the creation of physics simulations where multiple bodies (such as three planets) have concurrent affects on one another. Such situations have the benefit of having an existing to model, and the ideal solution is a compromise between accuracy and computational intensity. In the case of abstract cause and effect relationships the slavish reproduction of reality is not a goal, and may actually be detrimental to the creation of innovative causal relationships. In such a situation it may be more ideal to favor the how people would interpret such systems to work; in other words, favoring the simulation of naïve physics over real physics.[12] For example; there is evidence to suggest that our minds organize events in part by attempting to formulate a singular 'actor' that carries primary responsibility for events (Pinker 219). The topic out of scope here; but one of great relevance and potential direction for future research.

In the specific case that the feedback look arose, my personal intention was to state that two items move together as one regardless of which is being dragged, and that each could still be affected independently by other inputs. In order to express such a contingent "tying" together without creating an implicit feedback loop it may be useful to make a distinction between relationships that are "mono-directional" and those that are "bi-directional"; modifying the definition of a manipulation to include a 'directional' quality that can itself be manipulated. However, I find this adds syntactical complexity where emergent complexity would be preferred. The downside of favoring emergent complexity is that designers may have an increased ability to unintentionally create statements with completely broken results.

Implementing this language required several compromises due to situations like this which may be too detailed for this discussion. One detail is relevant however. In the implementation, change in values external to the artifact, such as mouse position, are always given precedence. In making a distinction between those values external to the artifact and those that are not I found it logical to describe time as a property external to the artifact and a potential driver of internal properties.

~~is always propagated through the system from those qualities that are closest to user input. The system itself handles this rule. A user (whether a designer or the user of a built system) shouldn't be able to break it accidentally. Many-to-one relationships may still be desirable, and I'll cover discuss it more in the next section "Drawing and multiples".~~

time

The results of these relationships intersect with time in a number of ways. Relative relationships in particular are implicitly temporal in that track change over time. In contrast, traditional computational expressions are executed once at a single moment of time — continuous change comes from their repeated use at specific intervals, such as 60 times a second. A program may control the intervals at which it runs (or unintentionally reduce them through difficult computations), it may also utilize an internal representation of time for manipulation (used to effect in games like Braid) but, like user input, the numeric values associated with the passage of time are external and may not be driven by other properties. Time may, however, serve to drive properties. A simple clock might begin with such a relationship.

Clock [2] seconds -> line rotation

While such a relationship may not utilize user input, these kinds of relationships may still be valuable in manipulations.

Change driven by time is ‘animation’. The distinction between this and change driven by user input may be unclear in places. For practical purposes, I’ll define animation here as change over time that does not require, nor respond, to continuous input.

It’s common for artifacts that respond to input with elaborate animation to be described as ‘more interactive’ and Interaction through this kind of ‘triggered animation’ is prevalent; One of the building blocks in Adobe’s Flash platform is the “Movie Clip”, a user created, self contained animation that can then be stopped, started, or sent to a specific frames.

As this is a low level model for interaction it should be able to describe common known interactions low complexity.

There also exists various code libraries (Like greensock.com’s TweenLite or jQuery) that allow for specific, computationally driven, animations by making a property of some object a function of time. Where as hand made timeline animation allows for individuals to craft animations, the computational approach is often used to create animations “on the fly”, in response to some input, and modified based on some variables in the system. For example, an animation could be coded that moves a shape to the cursor each time the user clicks. This is not possible via hand made animation. ~~Each animation would take the same amount of time and utilize the same kind of slow/fast/slow movement regardless of where the box begins or ends.~~

These libraries are noteworthy in that the required information for constructing an animation is similar to that of a manipulation. Both require a target object with a quantifiable property. Both take a specific length of time. The rage of values a property goes through over time may not be linear, but described by a mathematical function. With manipulation the current value of the modified property is dependent on the a user controlled value. In these computational animations time is the contingent value.

In this sense, a coded animation of the kind mentioned could be described as a kind of pre-recorded or pre-made manipulation.

How this might fit in a perceptual context is not entirely clear to me. I assume that the perception of a moving entity under the control of a user is both perceptually distinct from the passive observation of the same movement, but yet related — just as the object's otherwise static visual characteristics will influence a user's perception. It may be that such a "stock" manipulation could be representative of an action so practiced that it can be executed without conscious attention, and thus no input is required for it to happen. Regardless, a more robust model of manipulation that takes into account such non-guided changes would be useful.

Using time as a driving property also presents another situation in which an object may be driven by two 'actors'. *relationships 08* is one experimentation where both time and user input can affect a series of boxes. In it the change in position of the right most block is driven directly by the change in time. The relationship is based on sine creating periodic motion. This box's position is in turn mapped to other boxes with increasingly offset latency and increasing reduction, until the left most block which is entirely unaffected. The left most block is then directly manipulable. It's position also drivers the other blocks. The result is similar to a jumping rope held by two people at opposite ends.

Another example where manipulations that intersect time is when the kinetic qualities of an object need manipulated. This involves the manipulation of a relationship's 'mapping' property.

mapping

Computationally, a relationship's mapping property is expressed as a mathematical function. Such functions can take an endless variety of forms with a endless variety of potential properties and result in a variety of behaviors. *Relationships 13* contains some simple examples. In it are five boxes each draggable but in subtly different ways.

The most generic mapping function is linear. It relates one input value to one output value. i.e. If the cursor's x position is ever 100, the mapped value is guaranteed to be 100; if the cursors's change in x position was 10, the change in the mapped value is guaranteed to be 10.
Linear equations are expressed mathematically as such:

```
f(x) = x;
```

A simple variation on this mapping would be to change the output value by multiplying it or adding to it:

```
f(x) = x+1;
f(x) = x/10;
```

Note that one form has an offset while the other has a multiplier. The forms are distinct. The range of potential linear equations is endless:

```
f(x) = 2*x+1;
f(x) = (x*x) + 2*x + 2;
f(x) = 3*x*x+2(x*x) + 3*x + 3;
...
```

This problem becomes more difficult (or interesting) when additional 'free' parameters are allowed to enter the equation. These equations are known as parametric equations as they describe a *range* of linear functions, just like parametric forms describe a range of form.

The range of equations including an offset, where the offset may be any number:

```
f(x,n) = x + n;
```

The range of equations including a multiplier, where the multiplier may be any number:

```
f(x,n) = x * n;
```

These kinds of multi-parameter equations are useful in that they may be used to describe generalized behavior such as: the pen's x position follows the mouse's x position, but becomes more erratic as the mouse y position increases:

example?

```
f(x,n) = x * random(n);
```

Parametric equations are often used in animation programs or code libraries for describing a kind of movement ("start slow, then speed up") while leaving the beginning, end, and duration of the movement variable. A simple parametric equation that produces a constant change over time takes the form of:

```
current value = start value + (end value - start value)*(elapsed time / duration);
```

While a single parametric equation can be transformed into endless forms of linear equations (in order to describe movements of various durations with various starting and end locations), different parametric equations still describe distinct sets of linear equations. Two different parametric equations will describe two distinct kinds of movements even if the movement's duration, starting and end points are the same. The animation created by the above equation gives the object a constant velocity and is generally considered stiff, unnatural, or robotic. In contrast, other types of parametric equations can be used to describe an animation that has the kind of 'slow-in, slow-out' movement that gives objects the appearance of mass. These different equations can not be transformed into one another however. By extension, animation libraries typically require that a specific parametric equation be specified when an animation is created, e.g. Quadratic, Quartic, Exponential, Bounce, Elastic, etc.

http://www.robertpenner.com/easing/penner_chapter7_tweening.pdf

The variety and non-overlapping nature of these equations that describe a potential mapping prevents the creation of a set of convenient, manipulable, ‘properties’ to manipulate. In this situation, the need for abstract mathematical expression seems unavoidable — less mapping be constrained to specific, predetermined formats.

Providing a catalog of equations is one approach and is utilized by some animation software such as Adobe Edge and in Flash (specifically with the use of a motion tween).

Other software, such as Adobe After Effects, creates a graphical representation of the function and allows for it to be manipulated as a bezier curve. I find this far more intuitive and flexible. This approach is far from the discreet symbolic notation however, and would require an editor to accept such visual input, or, automatically generate the mathematical description of such a curve.

example

meta-manipulation

As mentioned, a manipulation has qualities of its own (target, latency, mapping). Thus relationships should have properties that themselves can be driven via another relationship. Here the the latency of the relationship between the mouse position and box position is driven by the mouse’s vertical position.

Δ mouse horizontal-position ->’ Δ box horizontal-position
Δ mouse vertical-position ->’’ Δ (->’) latency

Alternatively expressed as:

Δ mouse vertical-position -> Δ (Δ mouse horizontal-position -> Δ box horizontal-position) latency

As discussed previously, in the event that a user’s perception of the target of a manipulation shifts, it is ambiguous as to whether this should be described as the change in target of a persistent manipulation, or the abandonment of one manipulation for another. Such changes may be encouraged by objective changes to the chain of causality. Noting these potential manipulations (such as “the box *or* the circle can be moved”) may be desirable. One approach is to notate them as above, and allow them to be alternately ‘activated’, specifically through the inclusion of an ‘active’ or a ‘suspended’ property.

mouse button-down -> (Δ mouse horizontal-position -> Δ box horizontal-position) active
mouse button-up -> (Δ mouse horizontal-position -> Δ circle horizontal-position) active

(I find absolute mappings to be most useful in these circumstances.)

In the above case it would be useful if the entities themselves had properties such as ‘pressed’ and ‘touched’ that can be used to drive the activation of mappings.

box pressed -> (Δ mouse horizontal-position -> Δ box horizontal-position) active
circle pressed -> (Δ mouse horizontal-position -> Δ circle horizontal-position) active

These kinds of properties are examples of multi-variate properties — properties that consist of the result of other simple values, in this case a set of calculations concerning the distance between a two dimensional point and the geometric bounds of the visual entity in question along with the binary 'mouse button down' value.

((cursor distance to bounds < 0) *and* (mouse down))

drives

(Δ mouse horizontal-position -> Δ circle horizontal-position) active

The use of such combinatorial statements allows for many to one relationships and should be supported by any implementation of the model.

Two examples of simple interaction built from such basic 'clauses' exist in the studies *clauses_01* and *clauses_02*.)

While this approach is functional, it should also be possible to have the target of a mapping itself be the target of a mapping. The formal expression of this is problematic however:

box pressed -> (mouse horizontal-position -> box horizontal-position) target

Does the above mean that when the box is pressed, all instances of the value true in the system will then be equivalent to the value of the horizontal mouse position? It is far from a useful or desirable statement.

~~*The ability to map a binary value like "pressed" or "not pressed" to an otherwise continuous value like the multiplier of a mapping function should also work in reverse. In relationships 12 a set of boxes is mapped to the mouse (after clicking), but the otherwise continuous value of the mouse position is restricted to multiples of the boxes height.*~~

I believe a proper formulation requires the use of an abstract entity that represents multiples and allows for their manipulation.

and use of their qualities.

sets

Where people can deal with quantities as a singular entity, computers — baring multi-threading — modify many pieces of data not at at once, but sequentially, and very quickly. Loops and recursion are the de-facto tools for describing these repeating processes. The ability to have a computer do such repetitive tasks is arguably their most powerful and distinct qualities as a tool or medium. However, understanding the consequences of a looped process is difficult when the loop may run tens of thousands of times over the course of a second. For example, it's easy to understand the result of taking a step, or ten steps, but predicting your location after 1500 steps may be challenging without calculations.

For some operations it may be more ideal if the language could provide a more intuitive method of dealing with quantities. Some design decisions, like using the mouse position to drive the rotation of 100 squares, should be accomplished with brevity and elegance. This requires a method for articulating and resolving selections, e.g. “All the squares”, “All the black squares”, “All the squares to the left.”. Alternatively, an ability to label, create, or set aside arbitrary selections for future manipulations may also be useful, and a potential necessity for dealing with the variable target of a manipulation.

Both approaches are utilized in common web development practices. Firstly, HTML tags can have a singular identifier, “id”, any number of group identifiers “class”, and a distinct structural relationship. (Content elements are typically contained within another element.) In turn, the language of Cascading Style Sheets (or CSS) can be used to define visual styles for many elements with brief, simple, but powerful statements. The following CSS code will cause any number of header elements in a document to be displayed in a bold typeface:

```
h1{ font-weight:bold; }
```

CSS also supports more advanced selection statements such as “The first child *element* of *all* of some type of element”. This allows for stylistic treatments like making the first line of a paragraph italic without explicitly defining those particular items. These selections may also be malleable; if the column width of the paragraph changes, the style rules will apply to those words *currently* in the first line, opposed to those words that were *initially* in the first line.

Using these selection abilities, Javascript libraries like jQuery can change visuals rules in response to actions taken by the user. For example, clicking an item in a list changes the visibility of a sublist; the net result being a basic drop down menu. jQuery can modify style properties of a single item or a collection of similar items without a change in syntax. The following code will change something to red whether that something is one element or a collection of one hundred elements.

```
$(“.header”).css(“color”,”red”);
```

There are other technologies that include similar abilities such as the ECMAScript extension E4X for working with XML documents and the relatively complex but powerful Regular Expression syntax for use in finding selections in a text such as “any number that is preceded by a bullet point”.

The power of sophisticated selection abilities are seen in user interfaces too. Many existing tools, particularly 3D modeling applications, provide a variety of approaches to dealing with large quantities of manipulable items, usually in respect to their geometric relationships. Modelers are often provided with ways of selecting continuous loops of polygon edges or the creation of weighted selections of points where specific relationships are mapped more or less tightly based on the a specific point’s distance from an initial click.

Such systems can be described by a ‘set’ entity that represents multiples and has qualities determined by its contents. Letting the mouse position drive multiple box positions could then be expressed as such:

```
      mouse x -> (box1, box2, box3) horizontal-position(s)
```

Sets might also have qualities that describe the relationships between their contents:

```
      mouse x -> (box1, box2) distance
```

More interestingly, a set might include a set of qualities that describe a selection, or sub set, such as ‘start of selection’ and ‘end of selection’. These kinds of properties can be found in the text layers of After Effects and can be extremely powerful. Consider a line of 10 characters, the last two selected by a cursor and subject to manipulation; this may be described as:

```
character-set is (a,b,c,d,e,f,g,h,i)
character-set start-selection is 1
character-set end-selection is 2
character-set bold is true
```

The last line, instead of affecting all the contents of the set, would only affect the first and second items. If the input value was driving the end-selection quality, the set of bold characters may be manipulated:

```
Δ mouse horizontal-position -> Δ  character-set end-selection
```

As the set stands in as a proxy for any relationships to its contents, such selection qualities could be used to manipulate the target of another relationships.

```
Δ mouse horizontal-position -> Δ character-set end-selection
Δ mouse vertical-position -> Δ character-set vertical-position
```

Here the entity that the horizontal mouse position affects is subject to the vertical position of the mouse, and thus the target of the manipulation is manipulable.

example?....

~~Further research into both the ways in which we perceive collections of entities and their spatial relationships, and computational methods of describing groups would be useful to discover both what is most useful, and where common sense descriptions translate ambiguously into computation. For example, the desire to change the rotation of a set of boxes might translate to either the rotation of each box, or the transformation of the boxes as if it were a single object.~~

existing functionality

The functional result of these relationships is similar to functionality offered (as supporting tools) in various animation suites. Adobe’s *Aftereffects* animation and video editing software provides a “pick whip” tool on each layer that allows one to be “parented” to another so that transformations of the parent layer affect the child as if it were contained in the parent's co-ordinate system. This tool can also be used to create absolute relationships between different properties of a layer so that one quality might be adjusted through the manipulation of another. Maxon's *Cinema 4D* (a 3D modeling and animation suite) supports the kind of relative relationships through the options "set driver" and "set driven" accessed by a right click on any numeric property. As useful as these tools are for animation, I believe an application for creating interactive artifacts built primarily on these kinds of tools would be an even greater boon.

~~As part of this project, a rough implementation of this formal language was created in Actionscript 3.~~

---

[9] These terms come from the similar functionality of Maxon's Cinema 4D.

[10] This is in contrast to 'assignment by value' where the data that the right hand variable points to is copied into the memory represented by the left hand variable. In such a case, b would still be equal to 1 in the example.

[11] However, there are languages such as lambda calculus (that I am only now familiarizing myself with) in which such statements may be more tenable.

[12] A view which may conflict with one of my later conclusions.

Conclusion

revisions

The more I look at it everything, the more I think that latency should be tossed in favor of a description of mapping that is implicitly temporal. offset latency would just be a kind of ‘delayed’ mapping. There could also be ‘loose’ mapping, or ‘erratic’ mapping.

I can’t quite formulate how this works yet.

What I referred to as latency may be better expanded and described as the “temporal space” of a manipulation; the period of time over which a manipulation happens. This space might be described in terms of duration; the time between the moment an entity’s change is understood to be the effect of input and when the the desired change is complete. Note that this moment of understanding may be subjective. A individual new to a complex game with a great deal happening on screen may not recognize that their inputs are responsible for various immediate changes on screen. Alternately, in a situation where objectively observable screen changes are not synchronized with input, it may be possible for a user to understand the visual ‘silence’ as a meaningful and understandable response to their input, thus constituting the beginning of a manipulation from a perceptual point of view. This can be seen specifically in iOS devices that employ hold-to-activate buttons.

In these cases there seems to be an additional dimension regarding input. Several manipulations with the same “duration” may each require either continual input, momentary input with an anticipated but delayed result, or a momentary input with potentially variable results that require continual attention. Further investigation may provide a way to distinguish between where a manipulation may “bend”, and where it may deviate so much as to qualify as a distinctly new manipulation. For example, if the amount of movement required to move a square changed gradually as it was moved, it might be perceived as a continuous but changing manipulation, whereas if it changed drastically and instantaneously it might be completely interruptive and necessitate some kind of mental re-evaluation.

The “temporal space” of a manipulation might be further divided to include a ‘middle’ and ‘end’. This would be similar to how Steve Swink in *Game Feel* describes the response to direct manipulation in the acoustic terms: attack, sustain, and decay (121). I have not experimented with these divisions, but they may be very helpful categorizations, particularly in regard to a where a manipulation’s end is marked.

Previously I mentioned how a the objective results of a manipulation may be ‘good enough’ to subjectively qualify as complete. This could be called the manipulation’s tolerance. This might apply just as well to variation in the manipulated value at any point during the manipulation; variation in the ending state simply variation at one point in time. As such tolerance could also be defined as: The maximum amount of variance in the value space allowed, subjectively, before a manipulation breaks down, is discarded, or revised in some fashion.

continuous reveal

The studies demonstrate that descriptions of an artifact’s manipulability in terms of simple causal relationships may engender experiences of use that are more sophisticated, emergent, and/or complex than those typically attributed to a manipulation. Whether the engendered experiences qualify, for a lay person, as manipulation, interaction, or something else is ambiguous, but because of this, underscores the model's potential use in describing more complex interaction. Moving forward requires some of this ambiguity to be resolved however.

The model's requirement of intent as part of manipulation may be seen as problematic in the context of the studies. Intent may only exist where a desired end state can be envisioned along with the actions necessary for producing it. The studies though are characterized by mildly unexpected behavior and have no explicit goal or use.

As the target of a manipulation and the observable quality being modified are subjective, it may be possible to describe an initial manipulation in terms of 'I intend for the composition to change'. In other words, the artifact as a whole is the target of the manipulation, a general quality such as composition the target property, with a desired result marked by an exceptionally high tolerance (acceptable variance in the end state). Through such manipulations the user's understanding of what actions have what consequences is refined and their understanding of the artifact is clarified; A process definitive of active encounters.

Intent has further implications in the computational domain. Manipulation, like interaction, arises form a subjective encounter with an artifact and may not be explicitly designed, only designed for. Thus intent can not, and should not, be expressed in the computational definition of the artifact. Instead, the potential for manipulation is expressed in terms of causal relationships. Crafting of an interactive artifact is thus distinct in that it involves crafting artificial causal relationships.

Baring utilitarian uses, an ideal artifact is then one where manipulation reveals causal structures that inform and/or promote future manipulations. "Sandbox" games like Grand theft Auto or Sim City are good examples of this. Such artifacts are often described as promoting user generated goals (Costikyan ??, Salen and Zimmerman ??) and are contrasted with games that provide explicit goals.

I believe this distinction misleading however. A system that allows for a set of activities necessarily promotes those activities within the system. This is particularly relevant in what Ian Bogost describes as persuasive games; Topical games that present an abstraction of a real-life system in which players can encounter simplified causal structures in order to better understand the real life counterpart — or to better understand the author's view of the real life counterpart. In these situations the causal structure is rhetorical (communicative and/or persuasive). Similarly, an artifact's causal structures should be capable of expressing their potential manipulation.

As a direct encounter is valuable only so far as it helps elucidate the artifact's causal structures, an ideal interactive artifact is, more specifically, one that continuously reveals these structures. This might be most easily accomplished by introducing new variations into the system's relationships. However, the nature of a system's causal structures is definitive of the artifact, changing them may constitute changing the artifact. In practical terms a game with a series of levels each containing modified rules might just as well be described as a series of similar games played sequentially. The counter argument being that the actions in one level (specifically, completing it) do provide new causal structures (progression to the next level).

In this sense, the studies are more interactive in that they express unknown, unexpected, or surprising causal structures, but less interactive in that they do not do so continuously.

future research

…

Needs to be MLA not chicago

Works Cited

Brathwaite, Brenda. "Built on a Foundation of Code – Game Edu Rant «Applied Game Design", n.d. http://bbrathwaite.wordpress.com/2011/03/01/built-on-a-foundation-of-code-game-edu-rant/.

Brooks, Frederick P. *The Mythical Man-month: Essays on Software Engineering*. Anniversary ed. Reading, Mass.: Addison-Wesley Pub. Co., 1995.

Buxton, William. *Sketching User Experience: Getting the Design Right and the Right Design*. San Francisco, CA: Morgan Kaufmann, 2007. http://www.loc.gov/catdir/toc/ecip074/2006036416.html.

Cooper, Alan. *The Inmates Are Running the Asylum*. Indianapolis, IN: Sams, 1999.

Crawford, Chris. *The Art of Interactive Design: a Euphonious and Illuminating Guide to Building Successful Software, June 2002*. San Francisco: No Starch Press, 2002. http://www.loc.gov/catdir/enhancements/fy0716/2002001562-d.html.

Cypher, Allen, and Daniel Conrad Halbert. *Watch What I Do: Programming by Demonstration*. Cambridge, Mass.: MIT Press, 1993.

Dourish, Paul. *Where the Action Is: The Foundations of Embodied Interaction*. Cambridge, Mass.: MIT Press, 2001. http://www.loc.gov/catdir/toc/fy036/2001030443.html.

Gingold. *Miniature Gardens & Magic Crayons*, n.d.

Gingold, Chaim. "Catastrophic Prototyping and Other Stories | Levitylab", n.d. http://www.levitylab.com/blog/2011/01/catastrophic-prototyping-and-other-stories/.

Lakoff, George, and Mark Johnson. *Metaphors We Live By*. Chicago: University of Chicago Press, 2003.

Lopes, Dominic. *A Philosophy of Computer Art*. London: Routledge, 2010. http://www.loc.gov/catdir/enhancements/fy1106/2009007750-b.html.

Löwgren. *Five Things I Believe About the Aesthetics of Interaction Design*, n.d.

Löwgren, Jonas, and Erik Stolterman. *Thoughtful Interaction Design: a Design Perspective on Information Technology*. Cambridge, Mass.: MIT Press, 2004. http://www.loc.gov/catdir/toc/fy052/2004049891.html.

McCullough, Malcolm. *Abstracting Craft: The Practiced Digital Hand*. Reprint. The MIT Press, 1998.

Mccullough, Malcolm. *Digital Ground: Architecture, Pervasive Computing, and Environmental Knowing*. The MIT Press, 2005.

Norman, Donald A. *Emotional Design: Why We Love (or Hate) Everyday Things*. New York: Basic Books, 2004. http://www.loc.gov/catdir/toc/ecip043/2003010123.html.

———. *The Invisible Computer: Why Good Products Can Fail, the Personal Computer Is so Complex, and Information Appliances Are the Solution*. Cambridge, Mass.: MIT Press, 1998.

Norman, Donald. *The Design of Everyday Things*. Basic Books, 2002.

Raskin, Jef. *The Humane Interface: New Directions for Designing Interactive Systems*. Reading, Mass.: Addison Wesley, 2000.

Reas, Casey, and Chandler McWilliams. *Form+code in Design, Art, and Architecture*. 1st ed. Design Briefs. New York: Princeton Architectural Press, 2010.

Saffer, Dan. *Designing for Interaction: Creating Innovative Applications and Devices*. 2nd ed. Voices That Matter. Berkeley, CA: New Riders, 2010.

Salen, Katie, and Eric Zimmerman. *Rules of Play: Game Design Fundamentals*. Cambridge, Mass.: MIT Press, 2003.

Sharp, Helen, Yvonne Rogers, and Jenny Preece. *Interaction Design: Beyond Human-computer Interaction*. 2nd ed. Chichester: Wiley, 2007. http://www.loc.gov/catdir/toc/ecip071/2006030649.html.

Shneiderman, Ben. *Direct Manipulation: A Step Beyond Programming Languages*, n.d.

Snyder, Carolyn. *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*. San Diego, CA: Morgan Kaufmann Pub., 2003. http://www.loc.gov/catdir/description/els031/2002115472.html.

Svan\aes, Dag. *Understanding Interactivity: Steps to a Phenomenology of Human-computer Ineraction*. Trondheim: Norges teknisk-naturvitenskapelige universitet, Institutt for datateknikk og informasjonsvitenskap, 2000. http://www.loc.gov/catdir/toc/fy0609/2001319087.html.

Ware, Colin. *Visual Thinking: For Design (Morgan Kaufmann Series in Interactive Technologies)*. Illustrated. Morgan Kaufmann, 2008.