Chapter 12

Drawing Tools

> "Sim City is a paint program for making cities" - Chaim Gingold, "Magic Crayons", GDC 2007

What does it mean to draw? And why are so many casual creators *drawing tools*? In this chapter, I examine casual creators that use *drawing* as their primary form of interaction, and *paths* as their central data model. There are hundreds of casual creators whose possibility space is "a path in space"; how do these creators manage to have such different ways of interaction, feedback, and visualization?

12.1 Drawing on different perspectives

Drawing can be defined as the action of making marks on a surface[118], a practice humans have been engaged in for tens of thousands of years. But something obvious is missing from this definition: *why* do we draw?

Is it to record information? Pick up any "how to draw" manual, and it will have advice on perspective, accurate shading, creating outlines and forms, that is: many ways to accurately render a scene, real or imagined. Implicit when we say "drawing" is the idea that the goal is to create representational images,

images that record or capture something accurately. To be good at drawing, a "draftsman" like Dürer or DaVinci, is to be good at "capturing" a likeness. This is only one reason to draw though: we also make marks because we enjoy the marks themselves, as in the beautiful lines of calligraphy that transcend the words that they record. The dot artworks of Yayoi Kusama record her own intentions, or in the case of her interactive dot-placing works (Figure 12.1, the intentions of the gallery visitors[230]. We also draw for the pleasure of the drawing motion, drawing lines in wet sand with a stick, in snow with a glove, or in the air with a sparkler. The marks of an artist like Jackson Pollock record his gesture and the random physics of scattering paint, as Rosenberg describes in contemporary history *The American Action Painters*.

> "At a certain moment the canvas began to appear to one American painter after another as an arena in which to act—rather than as a space in which to reproduce, re-design, analyze or "express" an object, actual or imagined. What was to go on the canvas was not a picture but an event.[..] The painter no longer approached his easel with an image in his mind; he went up to it with material in his hand to do something to that other piece of material in front of him. **The image would be the result of this encounter.**"[326], (emphasis mine)

Archaeologist and art historian Whitney Davis considered it plausible that the art of mark-making, and thus drawing, arose from a gesture that left a record of itself. If true, the first 'paintings' were, like the Action Painters, representations of a gesture itself rather than a symbol to represent an external signifiers:

> In making a mark for the very first time, one could notice its visual or kinesthetic similarity to the depictive gesture. In the most literal version of this scenario, a depictive gesture accidentally leaves a trace in a permanent mark; here, "freezing" is occurring at the very moment of making both gesture and mark.(Whitney, 1993)

So if our experience of drawing can privilege any intention (recording a scene, making an aesthetic image, or leaving traces of a gesture) so too can our drawing

tools. When we build digital *drawing tools*, the distinction can be forgotten, and many tools are uncritically built to optimize for the control that a good draftsman would need to express their expertise. Precision, control, predictability are critical to digital tools with this goal, as well as in physical drawing tools like pens and pencils. It is easy to *assume* that representation and communication are essential elements of drawing, but they need not be. Reexamining this assumption allows us to find diverse drawing practices (in historical practice as well as digital systems) where this is not the case. From a casual creators perspective, we can consider *all* of the ways to experience drawing, and for each one, imagine a variety of tools that would support those experiences.

Pleasurable or meditative drawing provides a new motivation. *Zentangle*[s247], a proprietary form of meditative drawing practice, is an illuminating example, as it very explicitly advertises itself as a practice, not a way to get good art. The Zentangle site [406] characterizes it as "fun and relaxing" with easy-to-perform actions that lead to unexpected results:

> "The Zentangle Method is an easy-to-learn, relaxing, and fun way to create beautiful images by drawing structured patterns. We call these patterns, tangles. You create tangles with combinations of dots, lines, simple curves, S-curves and orbs. These simple shapes are the "Elemental Strokes" in all Zentangle art. These patterns are drawn on small pieces of paper called "tiles." We call them tiles because you can assemble them into mosaics. Zentangle art is non-representational and unplanned so you can focus on each stroke and not worry about the result. There is no up or down to Zentangle art. If fact, you can most easily create Zentangle art by rotating your tile as you tangle – always keeping your hand in a relaxed position. You don't need to know what a tangle is going to look like to draw it. You just need to know the steps. The result is a delightful surprise. As you use the Zentangle Method to create beautiful images, you likely will enjoy increased focus, creativity, self-confidence and an increased sense well-being."[407]

It also describes zentangling as ceremonial and timeless, citing the history and

meditative quality of repetitive mark-making, and mentions many psychological and personal benefits to the practice. Very little mention is made of the end products. The goal is not to produce better figurative or communicative images, but to make a better (or happier or calmer) person.

Figure 12.1: Interactive mark-making installations are popular in museums. These provide a wonderful space of casual creator examples, as they have to balance *some* creative expression on the part of the participant with automation or structure that creates a cohesive whole for the museum space. Yayoi Kusama's "Obliteration Room"[s151][230], Joseph L Griffith's "Drawing Machine 1"[s53][170], Karina Smigla-Bobinski's "ADA"[s3][171], Ryan Feddersen's "Coyote Now"[s37] coloring book mural at the Santa Cruz MAH[134].

12.1.1 Mark-making

The human practice of mark-making goes as far back in recorded history as recorded history does, for when we record history, we do so by making marks, and by making them permanent. Indentations on a clay tablet, were baked into permanence in ancient Assyria;the knotwork record-keeping of the Incan empire took momentary intentions and set them into permanence. Oak galls, lumps of misshapen wood formed when a wasp buries its eggs in a tree limb, are boiled, guided with capillary action down the tube of a split feather, and indelibly stain parchment and paper with the marks of the holy books and legal documents of Medieval and Enlightenment Europe. There are less-permanent marks, too: the sand mandalas of Buddhism, brushed away after creation, or the shape of footprints on a beach. We make marks with intentionally and without intention, permanently or ephemerally, for purposes serious and frivolous, in old clay and in digital storage. And of course, we make marks for art.

Why examine the "mark" as an individual entity, as this chapter does? For several reasons. We can analyze the mark by the properties that define it as a mark: its temporal permanence, its limited spatial existence, the necessarily linear nature of the gesture that created it. These properties will form an ontology of "mark-ness", and we can use these to understand the different ways that marks can be represented digitally, and knowledge of the diversity of digital representations will illuminate their diverse affordances. *How we represent a mark in code defines how we can interact with it.*

Putting attention onto "mark-making" also allows us to shift the focus from the mark, to the act of making it. What tool makes a mark? How is a tool held? What does it mean to "hold" a tool, when we are speaking digitally? What role do hands play in mark-making, and what are their affordances? What is a "gesture"

and how does it relate to drawing? If you can make a gesture without hands, can you also draw without hands? And this is the final purpose of this chapter on mark-making: by documenting what we know to be true of mark-making, we can discover the boundaries of our current assumptions. In this chapter you will see systems that make marks without gesture, without hands, without humans, and without a surface to make a mark on.

Anatomy of a mark

How can we represent a mark digitally? In the physical world, a mark may be made from presence (ink saturating paper), absence (the wax preventing ink from saturating paper, e.g. in batik), or difference (the height-change of carved lines in clay or sand). For now, let us consider the canonical stroke to be the mark of a ballpoint pen. This mark has several interesting properties:

- *It has a beginning.* There is a moment at which the pen nib hit the paper and *began* the mark.
- *It is continuous.* There is some span of time or distance in which the pen was not removed from the paper.
- *It is linear.* Time and space are linked, and over the course of some time, the pen has moved a distance, possibly changing direction, pressure, or width, but not jumping across or through the paper, or into the next room. Each point on the path it leaves is linked with a time, a single point, and a set of properties at that time and place.
- *It has an end.* At some point the stroke ended and the pen lifted from the paper. Our stroke exists for an interval of time, and according to the previous bullet point, also for an interval of spatial positions and fixed distance.

We can represent the position as a function of time or of distance (if the speed changed). We can also represent each of the changing attributes of the stroke (pressure, opacity, width, etc) as a function of the time (or distance).

This is our basic anatomy of a mark. There are several ways of representing marks like this digitally, with different affordances, as I found cataloged by early graphics pioneer (and Pixar co-founder) Alvy Ray Smith in his 1995 memo "Varieties of Digital Painting". [381]. Smith's memos on computer graphics were among very few ontological analyses of digital mark-making that I've found, and I was captivated by the idea that we could catalog all the ways that computers record and use strokes (and discover new ones).

His ontology includes two painting types:

- **discrete** painting, where the system records an approximation of marks by writing pixels in a pixel buffer. This system often uses a **read-modify-write** loop when adding marks-on-top-of-marks
- **continuous** painting, what is often called "vector graphics". A mark is a "continuous stroke defined by the spline", which must be rendered to be used as a painting operation, but can be manipulated independently (scaled, e.g.)

It also includes two accretion methods: **non-maxing** (writing directly to the buffer) and **maxing** (writing to an intermediate buffer for certain threshold-checking calculations, before then writing to the buffer). He also reminds that "Any imaging operation can be performed under the brush here, not just compositing". In current drawing tools, *Photoshop*[s168] has brush strokes that can blur underlying pixels, smudge them along the stroke, move backwards in the history of the buffer, or even fill them with machine-learned pattern matching from surrounding pixels[46].

The CS6 version of Illustrator (a vector-based illustration tool) has a robust brush-creation tool, and demonstrates a variety of brush types in its creation options:

- calligraphic: the angle, roundness and size along this continuous stroke can each be determined by different inputs (random, pressure, stylus wheel, tilt, bearing, and rotation)
- scatter: stamps some selected source object along the stroke, with size, spacing, scatter and rotation determined by inputs (similar to the calligraphic brush)
- art: a selected object is stretched continuously along the stroke
- pattern: like the art brush, the source object is deformed along the stroke, but the stroke is chopped into short sections so the pattern repeats. Can also have a custom head, tail, and corner version of the source object.
- bristle: creates multiple offset profiles of the original stroke, and creates continuous semi-transparent sub-strokes from them.

Based on the many brush examples I've seen[1], I propose adding to Alvy's ontology four ways of *shaping* strokes:

- **stamps**: "stamps" a source object along the stroke, without deforming it.
- **ribbons**: creates a geometric shape from two splines offset from the original stroke, and two splines for the two end caps. Some object (or sequence of objects) is distorted to match that object

[1]Based on brushes in *Illustrator*, *Tiltbrush*, *Oculus Medium*, and all the other art tools in this chapter.

- **extrusions**: a profile shape is extruded along the curve, creating a 3D tube.

- **diffusions**: the original stroke is multiplied and offset, or branches outwards, or breaks into simulated particles. Each substroke can rendered be as stamps, ribbons, or extrusions. Examples: the tree, drip, and loopy brushes in *Kid Pix*, the *Illustrator* bristle brush category. Can become **confusion** brushes when the generativity overwhelms the ability to control it.

Diffusion brushes are an interesting category, as they inherently involve a lack of control, and often use some form of generativity in the diffusion (like the Kid Pix Lindenmayer-system trees). You may remember that the casual creator tradeoff is "lose control to gain power", so these brushes are fruitful tools for casual creator experiences.

It is also important to consider how strokes are stored and whether or not they are *accessible* after having been recorded. For a traditional digital paint program like MS Paint, the canvas is a simple pixel buffer, and most marks just write a color into an area of the buffer, or start a function that recursively writes color into expanding areas until a stop condition is reached. The buffer is never erased, only added to. An Illustrator graphic has no persistent buffer as its canvas, but instead has a scene graph. If the scene graph is changed (a mark is added, a color is changed) the scene graph renders itself onto a temporary buffer and onto the screen.

These two approaches are **additive** and **functional rendering**. In additive rendering, there's a buffer that gets added to and evolved each time, based on the previous state of the buffer. Nothing is ever discarded, it's just built over.

Functional rendering is rendering without a buffer memory. Each time it renders the whole finished product from scratch, doing whatever subrendering or compositing it needs to do along the way.

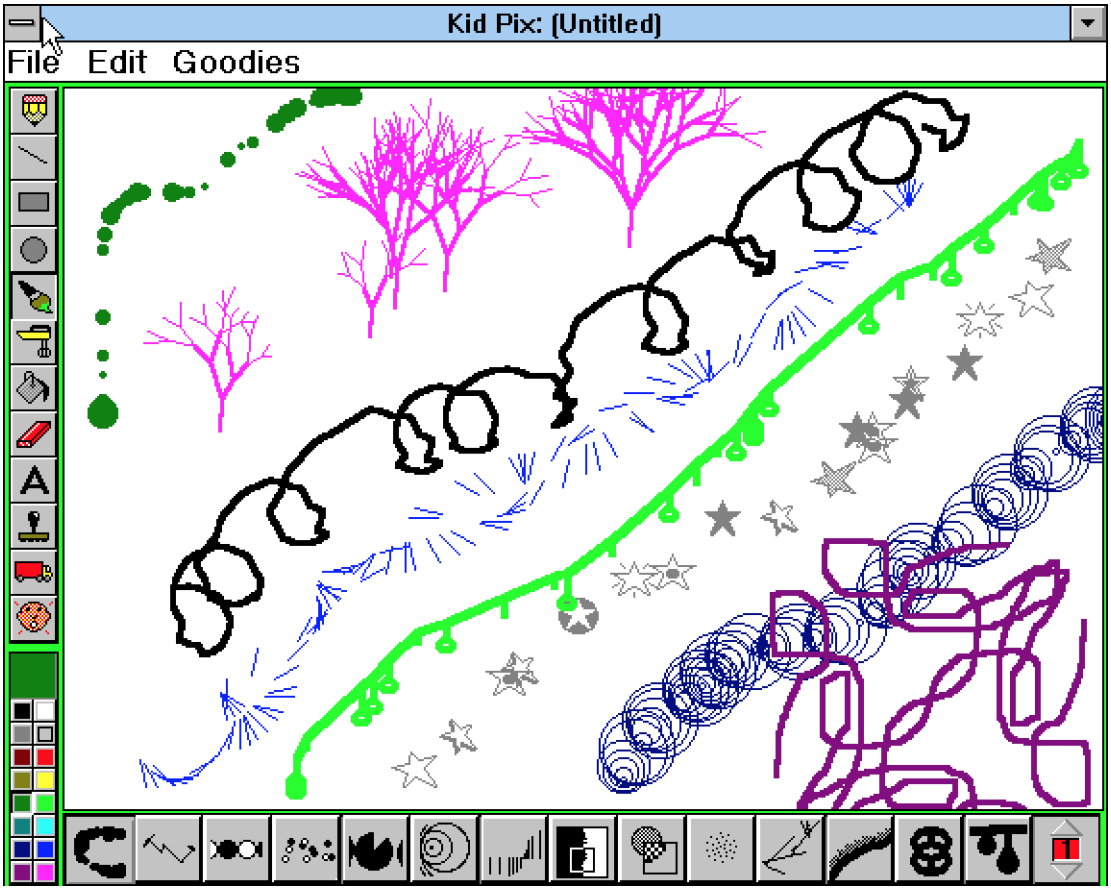

Figure 12.2: Diffusion brushes in *Kid Pix*.

The boundary between these two is surprisingly flexible. In a buffer-based drawing engine like Processing or HTML canvas, I may have a scene graph in memory that I want to redraw each frame. I can "erase" the buffer before drawing each time (setting each buffer pixel to a background color), and thus achieve functional rendering. OR I can choose not to clear the buffer, and each functional rendering will be made on the top of the previous rendering, additively. This technique leaves the past trails of a moving object visible, which can be good for inexpensive trail-rendering, notably used in the Windows Solitaire win animation and the Microsoft 3D Pipes screensaver. There's also an interesting middle path, where the previous rendering is partially obscured. In my work, I often draw a

partially transparent rectangle on top of the previous rendering, leaving the trails visible, but fading out over time in an attractively-veiled way.

Two competing VR drawing apps, *Oculus Medium*[s153] and *Google Tiltbrush*[s235] seem similar at first, with identical inputs, and both using a gestural curve model to record their marks. But under the surface, one is an additive and the other is a functional rendering, and this changes their interaction. *Oculus Medium* uses voxels, a 3D version of the pixel buffer. The user can "mark" more voxels into space, or write over previous voxels (or write over them with blank space to erase). But one can't go back and edit an earlier mark, other than by undoing to the point it was made. TiltBrush is fully functional, where each stroke is stored as a vector, then "rendered" into the scene graph (which is later rendered onto the screen in 3D). This allows modification of previous strokes, and more interesting ways of rendering strokes as animated curves or particle effects or patterned ribbons.

Tiltbrush and *Oculus Medium* are both intended (if unintentionally) as tools for good artists (note how their promotional materials show famous Disney animators and illustrators using them[258]). They want to be flexible enough for anyone to build anything, so they kept their rendering very directly related to the input.

Non-stroke mark-making

Strokes are not the only way that a mark can be made. Consider "regions" of a surface. These may be defined by a stroke, as Illustrator vector objects are, or they may encompass the entire surface. If the surface is a bitmap, a grid of pixel data, then some function can be run on each pixel in the region. The flood-fill operation is found in *Microsoft Paint*[s141] and *KidPix*[s113], but also in many other drawing tools. This operation take a single gesture, a click inside a region, and recursively applies a color to the pixel then to its uncolored neighboring pixels,

until all reachable pixels in the region are colored. This action is a template for many other actions, such as Photoshop's Lasso selection tool, which uses the flood-fill algorithm to recursively add similar pixels to a selection group as long as they are a similar color to their neighbor. The resulting "mark" is a selected region, to which other operations can be applied.

Regions

One example of a non-stroke "mark-making" interaction is the act of designating a flood-fill for a region. Several forms of casual creator are based around the selection (or application) of region-based flood fills. Embroidery patterns, cross-stitch patterns, and paint-by-numbers all consist of a preexisting set of regions, with proscribed fill colors. The user's activity is applying "strokes" to fill each region using paint or thread. The user makes no creative choices in the activity, but I would still considered it a creative activity because the user experiences the pleasures of seeing the picture develop from their own labor.

Coloring books add a dimension of choice: the regions are still fixed, but the user selects their own colors for each fill, then makes the pencil marks to fill those colors in. The user finds enjoyment in seeing the emergent results of the color applications, as the colors interact with each other visually, sometimes in unexpected ways. Digital coloring books remove much of the labor pleasure, as tapping an iPad screen, or worse, clicking a mouse, is a less physically-pleasurable feeling than the soothing frisson of dragging a soft pencil across paper, but the user still feels the pleasures of emergence and fulfillment.

Some activities do not start the user with an existing set of regions. Quilters, for example, may choose to recreate an existing pattern, to design their own pattern from existing components, or to create something with their own choice of

shapes. In the *Zentangle* activity[s247], the user is encouraged to freely draw looping lines across each other. These self-intersecting strokes create a pattern of regions, like a coloring book that emerges from the semi-directed actions of the user. The user then outlines each region and fills it with a different repeating fill pattern, of their own design, or of those suggested by other users. The finished patches of contrasting colors, lines, patterns, and dots create a shimmering patchwork effect that can be attractive without any particular skill on the user's part. *Zentangle* is an example of a casual creator *not* in a digital space: the users give up the fussiness and control of hand-drawing a particular image, instead following a set of abstract procedures in order to get a surprising and emergent design. Although the user could at any point make the choice to start drawing as they please, they constrain their choices to very few at any moment: where to loop the lines, which pattern and color to pick, which piece to fill in next, and how to continue or vary the pattern they are currently drawing. This limited choice paired with continuous action encourages a flow state, and indeed, this is the goal of the *Zentangle* experience.

There don't yet seem to be many casual creators that make interesting use of floodfill. Click-to-color apps are common, and work much like the color-by-number apps discussed in chapter 10. *Photoshop*[s168] has content-aware fill, which has potential for casual use, but is technologically advanced.[46] A few systems also use machine-learning to automatically "color" in images, like *Edges2Cats*[s61] and *Neural Doodles*[s148](which will be discussed further in chapter 22) which take user-provided line drawings and colored regions and fill them with machine-learned cats and impressionist paintings, respectively. But this process is not interactive, except for the creation of the line drawing itself.

Rendering

It is possible to consider 3D rendering to be a form of mark-making. When a 3D scene is ray-traced, are "marks" being made? An image is certainly appearing. It *is* useful to consider it as a form of mark-making, because when it is juxtaposed with interactivity, there are interesting designs that emerge. Though I have never seen this done, consider a ray-tracing app that allows the user to draw into a pixel buffer, but for each pixel they fill, the ray-tracing algorithm traces that pixel's light path, and colors the stroke with the 3D rendering of what is behind it, making a digital coloring book where you could scribble in the wireframe outlines with automatically shaded colors.

There are also other per-pixel algorithms that are worth considering as mark-making, such as Google's DeepDream. In DeepDream[278], for each patch of pixels in an image, a pre-trained neural network is used to optimized that patch towards being recognized as a dog or a tree, or other label. Like the raytracing example, we could imagine a drawing tool that allows the user to "draw" into the image, but each stroke written into the image would be an iterative optimization of the path's pixels towards some heuristic. The user would be painting with a "doggification" brush, with each successive stroke applying yet another optimization pass on the underlying pixels.

12.2 Modes of drawing in casual creators

Now that we have a robust ontology of all the things that can compose a possibility space of marks (thus creating a drawing tool), we can examine the ways in which we interact with drawing tools. I propose four new **modes of interactive drawing** in an autotelic experience, to provide alternatives to the

traditional one of representational sketching.

12.2.1 Conformance

The first mode is **conformance**. In this mode, the user is making marks, but they are making marks according to a pre-established goal. Success at this activity is measured not by creativity, but by conformance to the guidelines. Paint-by-number apps[s159], described in chapter 10 are one example of conformance "drawing", but there are also many learn-to-draw apps in the app store, which have a similar conformance play style, where the user traces over lines in a drawing. Handwriting and calligraphy apps such as *Calligraphy Handbook*[s28], *LazyDog calligraphy and cursive writing practice*[s120] and *HelloChinese's*[s92] character-drawing mode fall into this category as well. Though it is rare in games, in the action-adventure game *Okami*[s154], the user collects "celestial brush techniques"[93], patterns which can be traced with the Wiimote to cast spells.

This mode is *not* inherently conducive to casual creativity, though it is often conducive to flow. The addition of small moments of creative control (such as picking brush styles and colors in *Calligraphy Handbook*) can bring these systems closer to a casual creativity experience. I include this category in the chapter because it illustrates how drawing *is not* inherently a creative activity, and that an app that has a "drawing mode" is not instantly a casual creator.

12.2.2 Operational

In **operational** drawing, the user draws *in order to control some action external to the drawing*. *Okami* actually contains at least one of these in the brush technique "Bloom", which can either activate a tree to bloom by drawing a circle, or to freely draw a path along which trees and flowers bloom, which can be used

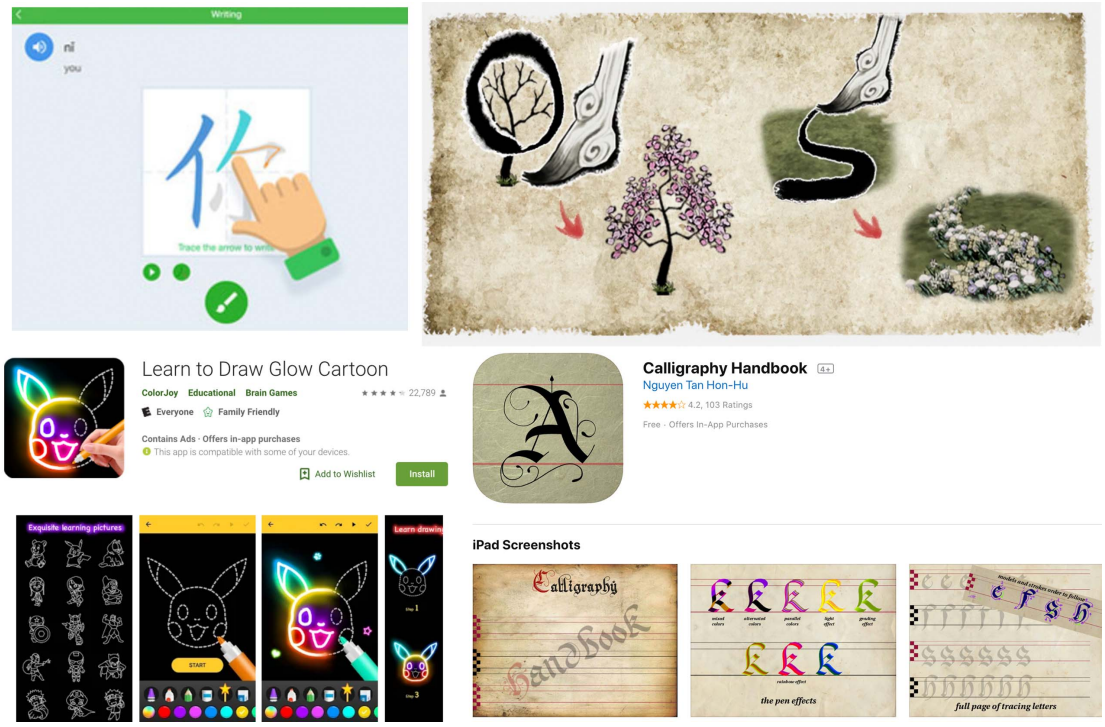

Figure 12.3: Conformance drawing in *HelloChinese*, *Okami*, *Calligraphy Handbook*, and *Learn to Draw Glow Cartoons*.

in combat to slow enemies. The Nintendo DS, with its stylus-controller sparked a boom in using "drawing" to control tradtional game movement. In *Kirby: Canvas Curse*[s114], the player draws rainbow-colored paths that participate in the system's in-game physics, allowing Kirby to slide or roll along them to traverse the level. While "creative" solutions are a part of the game, it never feels like the player is *creating an artifact* as much as they are *controlling a game character*. There are however, clear casual creators that fall into this category.

Line Rider[s124], a 2006 online Flash game by then-student Boštjan Čadež,[451] is a casual creator where the user draws lines, which are then used as a sledding hill by a tiny physics simulated person on a sled[2]. Users began creating intricate and wildly creative levels where the sledder's fall would synchronize with

[2]With a physics simulated scarf.

music[418], staging user-led contests[100] and collecting and archiving levels[183]. *Draw Your Game*[s54] is a creator much like *Line Rider*, except that users draw levels on paper (with black-for-platforms, red-for-danger, and blue-for-movable-objects pens) which can then be photographed to load into game levels, and then traversed interactively by a virtual character.

Operational drawing tools may or may not be casual creators. If the goal is to operationalize drawing in order to get past something, then they are not. If, on the other hand, the drawings are used to *create* something of some permanence that the user would look back at (or save, or share), then they are casual creators.

12.2.3 Accumulative

Recall above where art historian Rosenberg says that Pollack's action-painting is "not a picture but an event", a "result of [an] encounter". A drawing system can be an **accumulative** record of actions or forces.[3] In this mode, there is an interesting set of forces operating on the world, and the movement caused by those forces leaves a visible record of markings behind.

All four of the interactive installations in the image at the beginning of this chapter are accumulative works. "Obliteration Room" shows the accumulation of human attention to a space (especially low-down, where children can reach). The rings of "Drawing Machine 1" are built up from the radial motion of the armature, but also marked by the imperfection of its intentional shakiness. The "ADA" creates clusters of directional strokes, as the users spin it. "Coyote Now" is layered with the meticulous coloring, childlike scribbles, political slogans and profanity of Santa Cruz, all overlaid on top of each other.

[3] I tried out a number of names for this category, including "trail-blazing", "erosion", and "force-map", but no English word captures "a record of forces over time". Probably there is a Norwegian word for "the directional patterns of footprints in snow that shows you the group dynamics of the people who have passed".

Figure 12.4: Lava flows, erosion, and footprints are a frozen record of forces and dynamic movement. Even if we can't "read" the geology, their shapes implicitly tell us about the directions and collisions that happened here (CC-licensed photos from Pxhere, Gina Dittmer, and Tari Noelani Mattox).

This mode works well with agent and particle based systems. In *Action Painting Pro*[s2], the player controls a character jumping around a traditional platformer level, however, as the player moves, they leave behind a cascade of paint particles, recording their traversal in a vivid smear of colors. Pickups in the level change the painting style and color, so the goal is to reach the exit, while also producing an interesting painting. A reviewer of the game describes their accidental painting, and we can hear the casual creator properties of surprise, pride, and ownership in the way they narrate it:

> "Yeah! I didn't intend to draw a perfect dragon (or sea serpent), it just turned out that way. If you pay close attention you'll notice that the body's two humps are the trail of a platform character jumping to the right. The head was created when I collected a spraycan or something and my green expulsions somehow crafted a perfect visage, complete with an eye that is very much a window to the reptilian (aquatic or otherwise) soul behind. It's beautiful."[377]

StrangeThink's *Joy Exhibition*[s107] is another game where users control streams of particles with a video-game trope. But instead of a platformer, *Joy Exhibition* borrows the interaction style of a first-person shooter, complete with paint guns

that are evocative of alien weapons. Instead of bullets, these guns fire paint sprays that react with the screen in different ways, changing color with distance or angle, smearing, blurring, or speckling the wall with different colors. There are no editable layers or undo button, the player is encouraged to just build up an accumulation of paint until they are satisfied. Accumulation-mode drawing tools usually are *not* implemented as functional rendering (or editing of components), instead they additively write and rewrite a buffer.

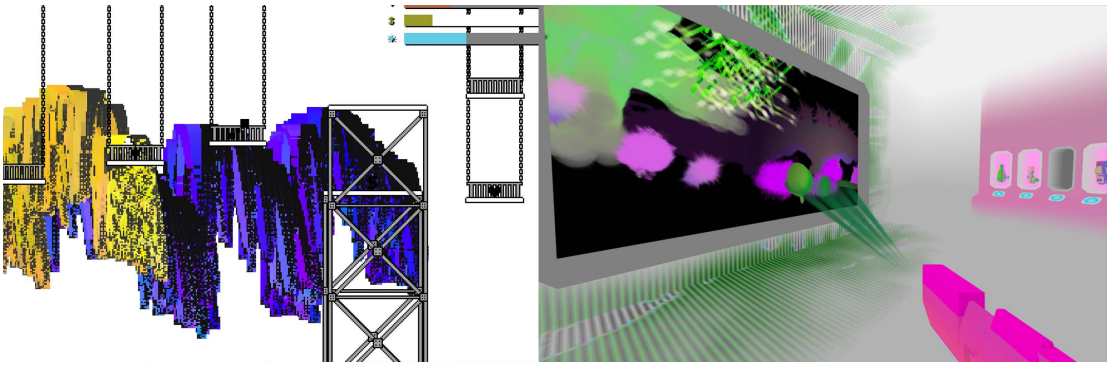

Figure 12.5: Action Painting Pro and Joy Exhibition.

Accumulation-mode drawing can also function synergistically with body-tracking systems, where we want to record force, gesture or body movement. One particularly wonderful and long-lived genre of this is the use of *still images* or moving images as *stamps*. Norman McLaren's 1968 dance film *Pas de Deux* uses multiple exposures to turn ephemeral dance moves into a record of spatial movementFigure 12.6[265]. When the Kinect came out, it was discovered, by artist and technologist Matt Bell, that it could be used to composite multiple 3D "still images" (voxels) into a single frame, which he used to create art videos[28]. DoubleFine's *Happy Action Theater* repackaged many of the experimental art Kinect scene's tools into a commercial Kinect party game, including using the 3D compositing trick to create "Future Booth", a time-travel themed photo-booth in

which users compose their photo not only spatially by temporally.[4]

These tools don't use paths, and so may be a fuzzy example of a "drawing tool". We can see this same technique used *along a drawn path* in *I/O Brush*[s104] [338], a 2004 project in which a "paintbrush" with webcam and position tracking can draw a stamped stroke. The stamps are sampled from the world by pointing the paintbrush at them and pressing a button to capture the image (as a still or video clip), like a paintbrush soaking up paint. The drawings, then, become collages of blinking eyes, ladybug patterns sampled from a purse, or animated motions of toys. Like the three accumulative cinematography/Kinect projects, I/O brush records motion, then combines that along a secondary motion of a brush stroke.

Figure 12.6: *Pax de Deux*, Matt Bell's Kinect experiments, and *Happy Action Theater* [178] [28] [265].

Body-tracking can be used to drive particle systems. In my own work, *Paintthrush*[s160] [67], I combine body-tracking, accumulative drawing, and particle systems into a casual creator. *Paintthrush* is a simulated world in which numerous birds are swooping around the space leaving behind brightly patterned semi-

[4] They don't cite Bell or any of the other Kinect artists in their GDC talk, but I have suspicions that they saw them.

persistent trails. The birds are attracted to the user's hands with a springy force, allowing the user to guide nearby birds with their fingers, or close their hands to increase the force and pull the birds into a tight swarm. The birds themselves paint into a 2D buffer behind the 3D scene. The intention behind this project was to make a casual creator where user's could *draw* indirectly with swarms of particle birds (very much a "diffusion" brush from the ontology above). The drawing is poorly-controlled, as the user indirectly directs the birds with their hands. It ended up feeling *too* indirectly controlled to be a satisfying art tool, though the birds were fun to interact with.

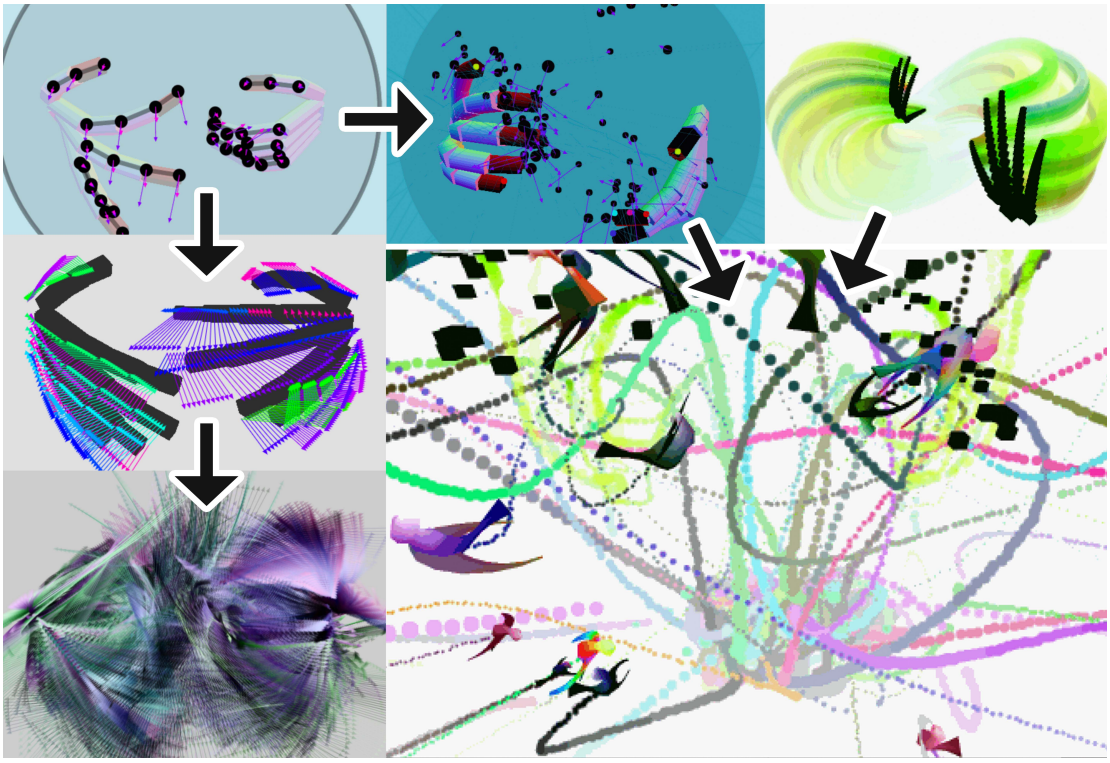

Figure 12.7: *Paintthrush* is a mixed particle-system/accumulative drawing tool. This image shows a number of prototypes that I went through, including some where there were no birds, but the acceleration vectors were drawn *accumulatively* onto the canvas, creating an appealingly shaggy record of hand motion.

12.2.4 Transformational

The final kind of drawing mode is **transformational**, where the strokes created by the user are somehow *transformed* or elaborated on by a generative system. The user is "drawing" in a traditionally-controlled mark-making sense, but the output is surprising, constrained, automated, or generative. Consider the *Spirograph* (a brand-name for a range of mathematical gear tools[243][s214]. I alway enjoy using the *Spirograph* toy as an example, as it is one of the strongest examples of non-digital casual creator; it is also a transformational drawing tool. With the Spirograph, users are in one way performing a very traditional drawing action: selecting a colored pen, moving it to leave a mark, and then picking it back up. But the Spirograph constrains their motion and guides it (by way of interlocking gears) into a more geometric, neat, and surprising path than the user could have made on their own. The user cannot draw the same possibility space of paths that they could freehand, but by trading that control, they get a great amount of augmentation and power, and a reliably good drawing. Their uncertain human gesture is *transformed* into a serenely mathematical one.

Kid Pix[s113], a 1989 (with many updates) computer drawing tool for children, shipped with many different brushes that the user's could draw with. A few were ordinary (discrete, closely-placed stamps to approximate a smooth curve, drawn additively into the buffer). But *Kid Pix* became famous for its many transformational brushes. These created Spirograph patterns, multiplied the stroke symmetrically, scattered multi-colored dots along the path, or came to life (dripping, growing trees, spiraling around the stroke). Craig Hickman, the creator, intentionally made these brushes as a rejection of traditional digital paint tools (and their lack of surprise)

> "In addition to "traditional" computer paint program tools like the brush, pencil, eraser, circle and rectangle, my program would include

> **tools that would be surprising and visually unusual.** Most computer paint programs were designed in a way that made them seem like second class citizens, always attempting to mimic traditional art making tools, but never doing a particularly good job of it. In fact, the computer can make marks in a variety of interesting ways that traditional tools can't. It can also do a good job of creating comic tools. This is where the "wacky brushes" came from."[193], (emphasis his)

Hickman used a variety of generative methods to create surprising brushes (*stamped*, *diffusion*, even some *confusion* brushes) even though they all end up written into a low-resolution black-and-white pixel buffer. Here in 2019, we have many more generative methods (and the computing power to run them), as well as different display options. Drawing tools are experiencing a renaissance in the current virtual reality boom. *Oculus Medium*, *Google Tiltbrush*, and Facebook's *Quill VR*[s183] are all tools for 3D drawing in a virtual environment. Quill and Tiltbrush both use a model of paths through space, which are rendered as textured 3D ribbons or extruded tubes. By repeating many strokes, users can build up planes and surfaces to create 3D shapes, or just make sketches in air. Neither of these tools are pushing themselves as casual creators[5], but *Tiltbrush* has a range of surprising and augmented "brushes" to paint with, including a few brushes of animated blinking scattered stamps, diffusion brushes of multiple intertwining threads, extruded neon tubes, and recently, a set of brushes that animate their settings according to music, so that any drawing with them becomes a music visualizer[281].

Transformational brushes can be used with many different generative pipelines, even very complex ones. One useful property of transformational drawing tools is that, if they update continuously while the user is drawing, the user has a

[5]I attended a marketing seminar for *Quill* in which the announcer derided the possible option to support casual users.

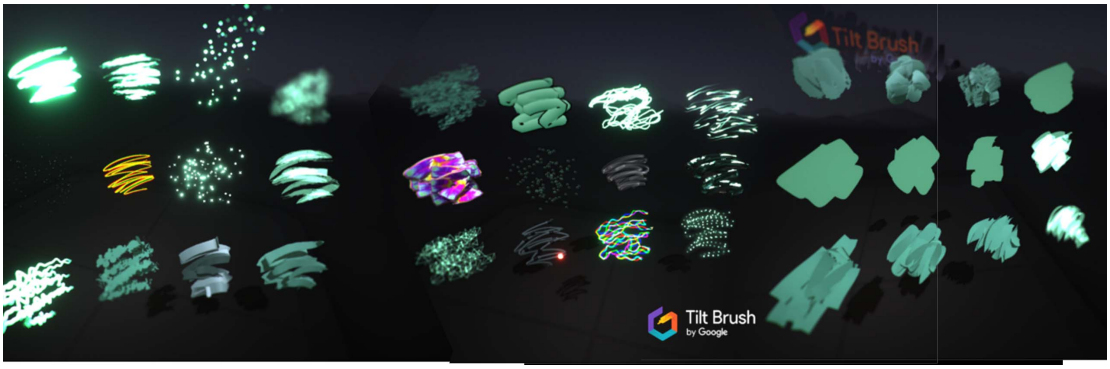

Figure 12.8: Brush types (from [239].

continuous grokloop telling them how their movements are shaping the strange new artifact that is being created. I discovered this property when I was working on a 3D drawing tool, *Pendant Maker*[s164], which had been commissioned by a (now-defunct) 3D printing startup to help casual users make 3D printable object without getting the art-paralysis that occurs when you ask people to "draw something". In *Pendant Maker*, the user draws curves into a small buffer, curves which are truncated to the last few dozen segments, are continually extruded into a 3D tube, are flanged and flared according to some sliders, and then cloned around a central axis to create a mandala-like shape. As a casual creator, it constrained the space to only 3D printable forms (watertight and limited overhangs) that could be quickly printed in a store. I *hadn't* anticipated that drawing with a continuous feedback loop would be so powerful. Users would doodle aimlessly, change direction, and *just keep drawing*. I had added an *undo* slider, but it was almost never used. People just went with the flow, feeling in control, but also empowered.

Like any casual creator, transformational drawing tools can play along the whole interaction-directness spectrum, from control to lively chaos. While *Pendant Maker* calls back to the Spirograph's enforced mathematical precision, my more recent work *Jazz Composition* (2018)[s105] is intended as a drawing tool that rebels against control. This tool lets the user draw into a gray background, ei-

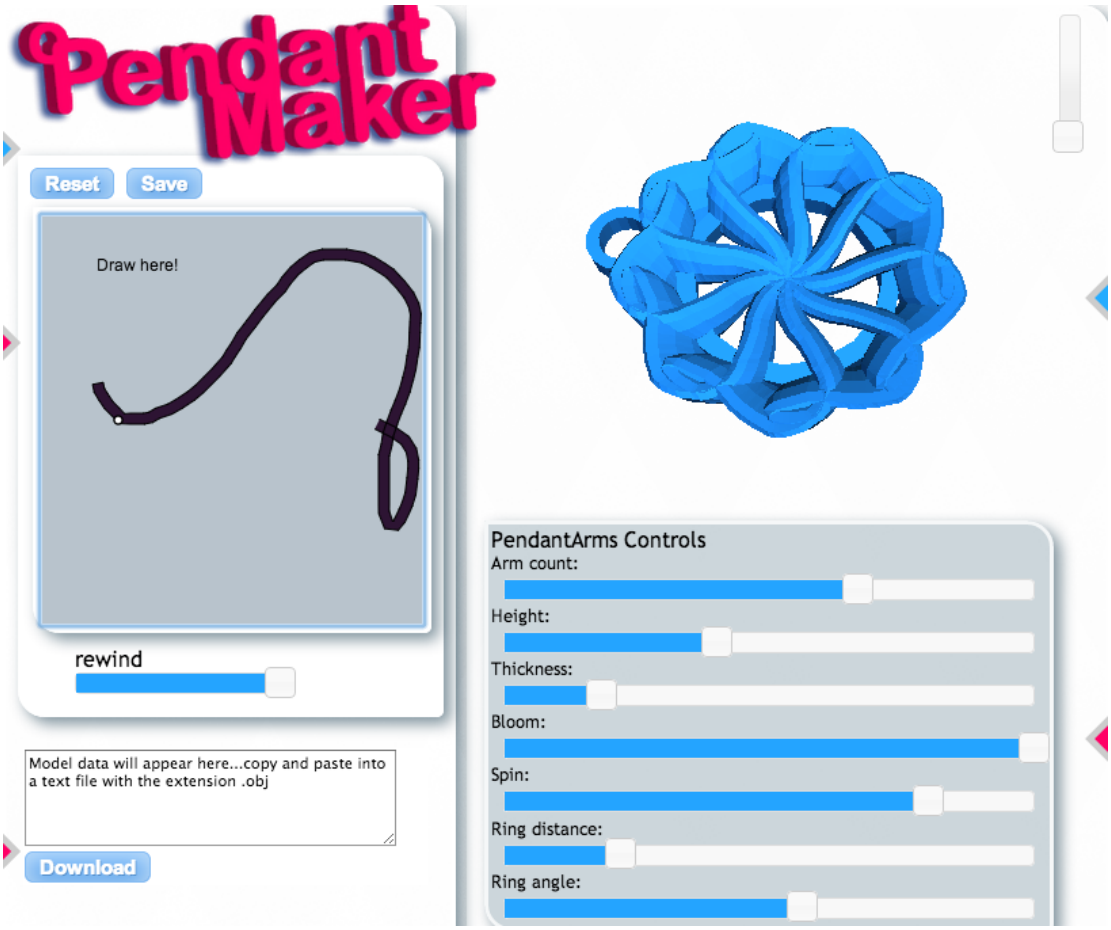

Figure 12.9: The *Pendant Maker* UI. It became quickly apparent that users didn't even look at the drawing buffer, they could draw while *only* looking at the extruding cloned 3D paths.

ther with a mouse, a touch interface, or with a Leap Motion hand-tracking device (starting and ending the stroke by pinching or releasing thumb and forefinger). When the user starts a new line, the system generates a new brush style, either a continuous line, a filled continuous ribbon, a filled region, scattered dots, or a multilayered diffusion brush made of many brush strokes offset from the original. The user only sees the tools when they start drawing, without control over it, but as they draw out the curve, it begins to pull away from where they have placed it. Behind the scenes, the stroke is a series of Bezier points, which are also attached

to simulated physics particles with a simple skeleton of springs which keep the line connected but also provided muscle twitches to animate it. So as they draw, the complexly-rendered stroke comes to life and twitches according to its own rhythm.[68]

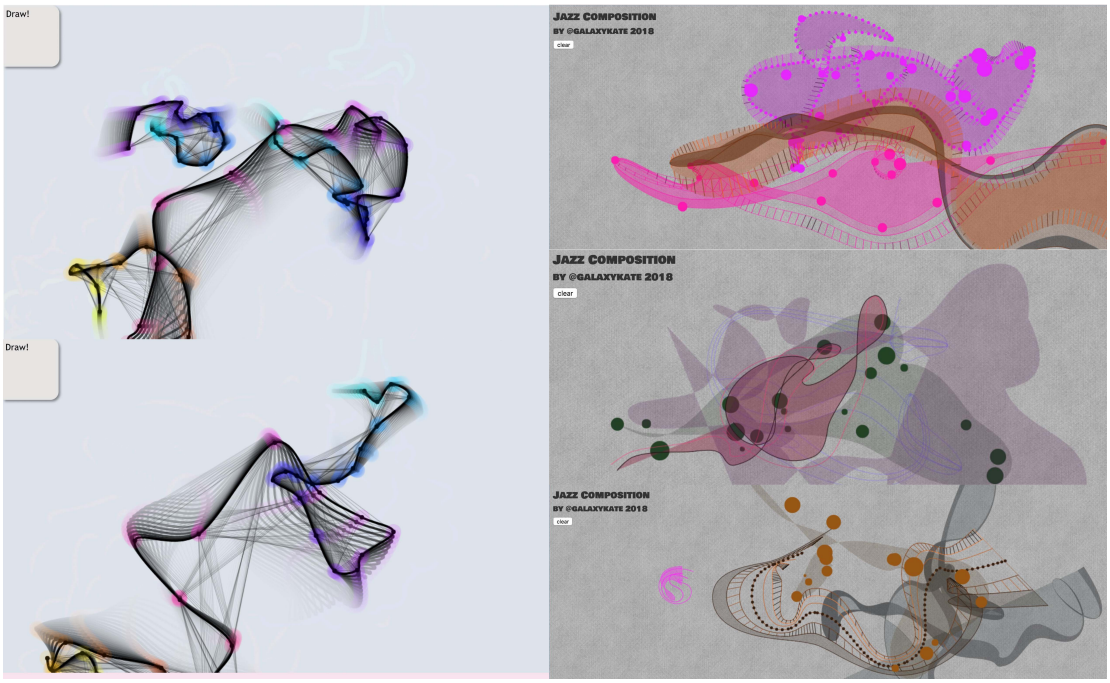

Figure 12.10: An earlier prototype of *Scribbles*[s197] (`http://www.galaxykate.com/apps/Prototypes/Scribbles/`), an earlier spring-particle drawing tool using accumulative drawing to show its motion over time (left), and several drawings from *Jazz Composition*(right).

Drawing tools are a very strong casual creator subcategory. Not every drawing tool is a casual creator: some have too little support (*Oculus Medium*) and some have too little creative emphasis (*Kirby Canvas Curse*). But for many other casual creator drawing tools, the interactive paradigm of "drawing" creates a strong and fast grokloop. The gulf of execution is neatly bridged by how natural it is to move our bodies in a drawing gesture (and how much internal proprioceptive feedback our bodies provide for that action). If a drawing tool provides continuous visible feedback of the stroke we are drawing (and for a drawing tool, it feels like a

betrayal not to have that), then the gulf of evaluation is crossed instantly (as long as we only care about aesthetics). Drawing tools almost always have a continuous groklook, which lets them get away with wild and alien logic in their generative algorithm. In the next chapter chapter 13, we will see a space of casual creators for which this is *not* the case, and how they deal with a painful slow grokloop while still eking out some casual creator experience.