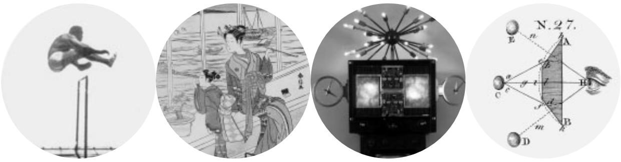
journal of visual culture



Because of the Pixels: On the History, Form, and Influence of MS Paint

Patrick Davison

Abstract

This article focuses on the history, formal properties, and cultural role played by MS Paint, a widely available graphics program that has been used to make countless internet memes. The author examines the technical limitations of the program, and the visible consequences produced by a mouse-driven bitmap-based graphics program. He uses MS Paint as a context to discuss two new pieces of graphics software – Fresh Paint and Rage Maker – each of which embody radically different orientations to the concept of remediation. These three approaches to understanding MS Paint help us to reconsider the culturally suppressed media that have nonetheless had significant effects on the visual culture of the contemporary participatory internet.

Keywords

computer graphics • digital aesthetics • Fresh Paint • internet memes • MS Paint • Rage Maker • software studies

Because of the Pixels

As Lisa Nakamura (2002: 1) wrote:

> 1995 was a turning point in the history of the Internet … In 1995 Netscape Navigator, the first widely popular graphical Web browser … initiated popular use of the Internet and, most importantly, heralded its transformation from a primarily textual one to an increasingly and irreversibly graphical one.

journal of visual culture [http://vcu.sagepub.com]
SAGE Publications (Los Angeles, London, New Delhi, Singapore and Washington DC)

Vol 13(3): 275–297 DOI 10.1177/1470412914544539

It is in the context of this increasingly *popular* and *graphical* internet that the term 'internet meme' has achieved its current cultural meaning. The term 'meme' was being used to describe the behavior of people on the internet as early as 1994, when Mike Godwin (1994) published his analysis of 'the Nazi-comparison meme' in *Wired* magazine. However, in the two decades since then, 'meme' has become specifically associated with the practice of producing, altering, posting, and sending countless digital images. The production and transmission of memes has become a contemporary visual vernacular, with numerous communities, both formal and ad hoc, using meme genres, such as LOLcats, Advice Animals, or Rage Comics, to communicate and socialize.

It can be tempting to focus a discussion of internet memes on the content of the images and the communicative function they perform. Why cats? To whom are these animals giving advice? What is the role of 'rage' in contemporary society? However, such an approach bypasses an important nuance: for every digital image identified as an internet meme, there exists some piece of software used to produce it. 'To understand media today', writes Lev Manovich (2013: 124), 'we need to *understand media software* – its genealogy (where it comes from), its anatomy (interfaces and operations), and its practical and theoretical effects' (emphasis in original). What follows in this article is an assembly of these three branches of knowledge for a piece of software that has escaped serious attention and yet has served as the primary medium for much meme-making: the bitmap-based graphics program known as MS Paint.

The name 'MS Paint' does not actually refer to a single piece of software. Rather, the title has emerged as shorthand for whichever bitmap-based graphics program is standard with the current Windows operating system (OS); this slot has been occupied by a number of different programs since the release of Windows 1.0 in 1985. While the Windows graphics program has been updated or overhauled almost as many times as Windows itself, most of these updates have done little to change its interface and underlying functionality as a program that constructs images as grids of pixels selected from a limited palette of colors and composed with a small set of mouse-directed tools. For over 20 years, from Windows 1.0 in 1985 to Windows Vista in 2007, Microsoft's standard graphics program offered a remarkably consistent user experience.[1] While more 'advanced' programs, such as Photoshop or CorelDRAW have swelled with 'must-have tools' or 'major feature enhancements' (Coreldraw.com, 2014) such as layers, filters, QR code generators and customizable interfaces, MS Paint has remained comparatively familiar and accessible since its initial incarnation in 1985.

If innovation and technical elegance were the only standards of a medium's historical or cultural merit, there would not be much to say about MS Paint. To borrow imagery from Ron Eglash (2008: 61), MS Paint has never been the 'cutting edge' of graphics software, but rather its 'trailing shadow'. Although, as Christina Lindsay (2003: 50) reminds us in her work on the TRS-80, 'the disappearance of a technology from mainstream public view is not necessarily the end of that technology.' Quite the contrary: just as MS Paint was slipping into technical obsolescence in 1995, it was simultaneously accompanying the 40 million copies of Windows 95 sold during the operating system's first year of release, mostly to laptop and home computing consumers who were also exploring the possibilities of a new class of commercial internet service providers, such as AOL and Compuserv (Burns, 1996: 25). The following five years would see a greater increase in the percentage of American adults using the internet than any other similar period – jumping from 14 percent in 1995 to 46 percent in 2000 (Fox and Rainie, 2014). The convergence of MS Paint's ubiquity, with the rise of Nakamura's 'increasingly and irreversibly graphical' internet, produced the circumstances under which MS Paint helped produce a visual, participatory, and online culture. This software was the graphics program most readily available and easy to use at the moment the internet took its graphical turn.

This article begins by assembling a genealogy of how MS Paint came to be the program it was in 1985 and what characteristics of this program persisted into the 21st century. There is currently no history of this program, and therefore this article engages in the modest effort of providing one. My own account begins in the decades before 1985, when early developments in computer graphics would prefigure the techniques used in MS Paint, set the performance standards against which the program would be evaluated, and establish the distinction between digital 'drawing' and 'painting'. From 1985 through 2007, the changes to MS Paint (or lack thereof) reflected the program's entanglement with Microsoft's marquee operating system and the various economic concerns that guided software development within the company.

The article continues with an argument that the recognizably rough visual character of images produced with MS Paint is not general or arbitrary, but rather a specific consequence of the way in which the program defines digital images and the tools it provides to interact with them. MS Paint is a bitmap-based program intended to be used with a mouse, and the visual character of this combination of software and hardware has over time influenced an MS Paint 'style'. Through a close examination of a notable example of a Rage Comic (taken from the Know Your Meme archive of a 4chan post from 2008), I discuss the specific visual character of MS Paint-produced images, and describe how the program produces an 'authentic computer aesthetics' (Goriunova and Shulgin, 2008: 113) rather than a simulation of traditional painting. This claim to authenticity is not a normative one. Rather, it is a description of how MS Paint makes visible the properties of the technologies it mediates (the bitmap, the mouse) over the technologies it symbolically *re*mediates (pigments, brushes).

The article concludes by turning from MS Paint's effect on the visual character of images to look at its influence on new two new pieces of graphics software: a commercial product produced by Microsoft as part of their release of Windows 8, named Fresh Paint, and the community-produced 'Rage Maker' that facilitates the production of Rage Comics for reddit.com users. By comparing these two programs in the context of MS Paint's history, I hope to illustrate how certain types of cultural production result not from a pursuit of technological 'progress' but from the lack of such desire. 'Widely endorsed accounts of contemporary media culture and media histories alike often tell only selected parts of the story' write Erkki Huhtamo and Jussi Parikka (2011: 3), and in the case of the internet these 'selected parts' tend to be the technical advances and the commercial successes. A historical focus on memes provides an antidote to this: a perspective on the parts of media and culture that are otherwise unrecognized. The influence of MS Paint on the memes and software of the last 20 years reveals that while MS Paint has so far been a prime example of a medium 'suppressed, neglected, and forgotten' (Huhtamo and Parikka, 2011: 3), it has nonetheless been one of the programs with the greatest influence on the visual culture of the internet.

A Short History of MS Paint

The technological decisions that shaped MS Paint, and, as I will later show, had visible impacts on the images produced with it, were 'often made on grounds other than technical limitations: for instance, on economic, political, ideological, or cultural grounds' (Tedre and Eglash, 2008: 97). Even as the first version of MS Paint was being designed, coded, and named, it was already pressed upon by diverse concerns within a wider historical complex of technological, economic, and social developments directed at using computer technology to create, process, and edit graphical data.

That Microsoft's graphics program was given the name 'Paint' might seem arbitrary, but by 1985 there was a formal and professionally acknowledged distinction between 'drawing' and 'painting' programs. This difference between 'drawing' and 'painting' has little to do with the art historical distinction between the two practices that lend their names (Nappi, 2013). Rather, for software produced in the early 1980s the terms 'drawing' and 'painting' designated the two primary methods used to digitally define images: geometry or bitmaps (see Figure 1).[2]

Drawing programs, like LisaDraw (1983) or Illustrator (1987), allowed the user to produce images by positioning and manipulating geometrically defined 'objects'. This was a technique first employed by Ivan Sutherland, with his 1963 program Sketchpad (Nappi, 2013). Using a light pen and a bank of physical toggle switches, Sketchpad users were able to specify points, vectors, and arcs on an invisible X,Y plane. Such digital images were extremely precise and therefore suitable for use in applications of engineering and design. They also required little storage space and could be easily scaled to any size or resolution.

Paint programs, like Windows Paint (1985) or MacPaint (1984), used a bitmap concept rather than geometrically defined objects. A bitmap is a two-dimensional presentation of the bits in a computer's memory. These bits could either be 'on' or 'off' and when these states were

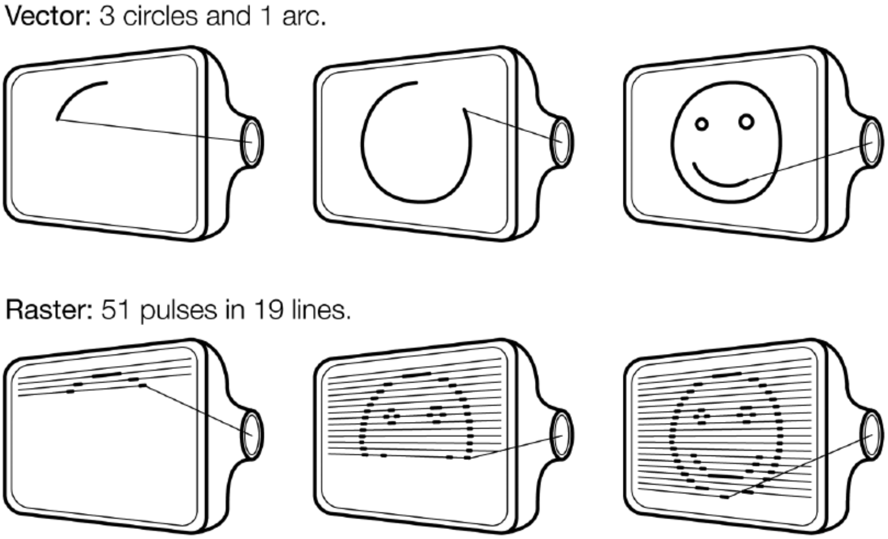

Vector: 3 circles and 1 arc.
Raster: 51 pulses in 19 lines.

Figure 1 The most common means of displaying personal computer output at the beginning of the 1980s was a cathode ray tube (CRT) monitor that used an electron beam to draw images on a phosphor-coated screen. The difference between vector and raster graphics software is mirrored by such a screen's physical operation: a vector display draws each geometric primitive in turn based on the image being processed; a raster display moves the electron beam in a strict pattern of rows, drawing individual points in a digital grid (Van Dam, 1984).

then rendered as black and white, respectively, the bitmap produced a visible image.[3] Lines in bitmaps were not stored as endpoint coordinates – instead all the bits along the line were simply 'turned on'. Dick Shoup developed one of the earliest bitmap-based systems, called SuperPaint, in1973 at Xerox PARC.[4] This system and others like it gave up the strict mathematical precision and efficient file size of geometric-based images, but gained the ability to record the smooth movement of different input devices – allowing for the direct trace of human hands in digital images.

Between 1975 and 1985, the difference between drawing and painting programs became a metonym for different types of computer users.[5] Drawing programs, based on pre-defined shapes and strict geometry, produced standardized and precise images and were therefore programmed for and marketed toward professionals and business applications. In contrast, painting programs allowed for the capture of free-form gesture. This produced a lack of precision but also an increased capacity for expression. Paint programs for personal computers, then, were programmed and billed as tools of personal expression for every member of the computer owning family.[6] 'Drawing' for professionals, ‘painting’ for everyone else – this was the accepted paradigm when the first version of MS Paint was released.

This first version of MS Paint, then named ‘Windows Paint’, was programmed by Dan McCabe and sold as part of Windows 1.0 beginning in November 1985 (see Figure 2).[7] Although Windows 1.0 came with a number of ‘utilities’ – including a clock, terminal, and the game Reversi – the marketing materials and press coverage of the time made special mention of Windows Paint and its word processing equivalent Windows Write as full applications instead of mere utility-like features. The decisions to not only include a graphics program, but to frame it as a desirable, fully developed program in its own right was motivated by Microsoft’s competition with Apple, whose Macintosh MacPaint program had been heavily featured in marketing and press coverage as a symbol for its user-friendly interface (Trower, 2014, personal communication).[8] Windows Paint and the other applications were intended to help users to ‘see the potential of the new environment’ and thus motivate sales (Trower, 2014, personal communication).

Windows Paint was black and white, contained 24 tools, and saved images with a (now defunct) .MSP extension. ‘The heart of all painting programs is their tools’, wrote Jeffrey Young (1984: 132) in his article in *Personal Computing*, ‘How to “Paint” with your Computer’: ‘while there are individual differences in the tools available, most of the programs use variations of the same theme.’ Windows Paint was no exception.[9] It included a ‘pencil’ that drew free-form lines, a ‘brush’ that drew in 24 ‘brush shapes and patterns’. There were ‘shape’ tools that allowed the user to click and drag to create rectangles, circles, and triangles. Windows Paint also provided two wholly original tools: a ‘bezier’ tool for creating smoothly curved lines, and an ‘isometric’ tool that drew lines constrained to three angles, producing a forced ¾ perspective (McCabe, 2014, personal communication).

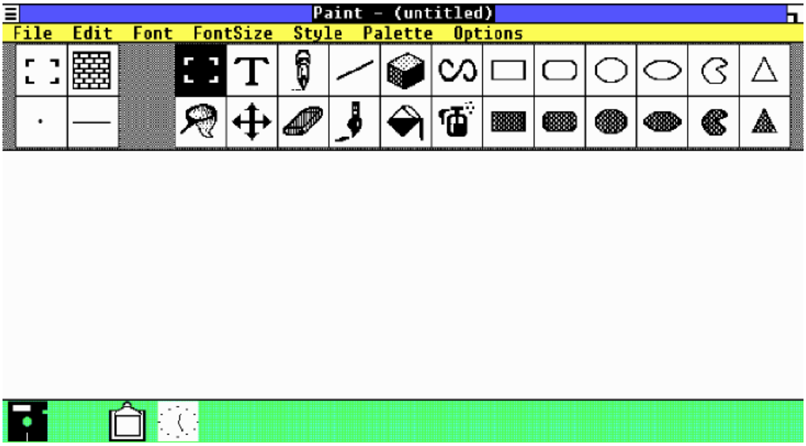


Figure 2 The interface of Windows Paint running on Windows 1.0. Reproduced with permission of Microsoft.

The release and sale of Windows 1.0 did not achieve the commercial success some had predicted (Watt, 1987). The system as a whole and Windows Paint had both been subject to a small development team and significant time constraints. When Windows 1.0 shipped in 1985, the team responsible already had plans for crucial improvements in the next version (Trower, 2014, personal communication). During this same time period, however, Microsoft and IBM (which sold the PC machines Windows was designed for) began to collaborate on the development of a new graphical operating system that would be known as Presentation Manager. This project was initially intended to supersede Windows (slated to cease development after version 2.0). Thus, between 1985 and 1987, Microsoft resources that could have been directed toward Windows 2.0 were diverted elsewhere. As a result, many of the less critical aspects of the Windows operating system (including the capabilities of Windows Paint) remained largely unchanged in Windows' second version (Trower, 2014, personal communication).

Despite the intended Microsoft–IBM collaboration, the project fell through, and Microsoft switched gears once again to develop a new and dramatically updated Windows. Microsoft adopted a new strategy for providing Windows 3.0's bundled applications. Rather than produce these accessory utilities with small in-house teams (as had been done with Windows 1.0), Microsoft contracted individual software vendors with successful products to provide limited-functionality versions of their own software as part of Windows 3.0 (Snodgrass, 2014, personal communication). Software developer ZSoft had been selling its own 16-color paint program – PC Paintbrush, programmed by Mark Zachmann – since 1985, and this became the code base for the next incarnation of MS Paint (Zachmann, 2014, personal communication).[10] With the 1990 release of Windows 3.0, Windows Paint was replaced by Microsoft Paintbrush.

Microsoft Paintbrush was closer to the program its code originated from – PC Paintbrush – than the program it was intended to replace – Windows Paint – but the overall functions and interface of Microsoft Paintbrush and Windows Paint were not so different (see Figure 3). The tool palette moved to the left edge of the screen, and the color palette stretched across the bottom. The 'brush' tool remained, the 'pencil' disappeared, the 'paint can' became a 'paint roller'. The shape tools fell from 12 to 8, and the bezier curve tool now required two control points (making 'S' curves possible). The largest change, however, was that the program now worked with color. Gone were the bitmapped checkerboard patterns of Windows Paint – replaced with hues produced by mixing 256 values each of red, green, and blue.[11]

Microsoft Paintbrush would remain the Windows standard graphics program for the next five years. The commercial difficulties for Windows in the second half of the 80s gave way to a period of great success in the first half of the 90s as the operating system shifted from an optional add-on to the default environment for PCs (Livingston, 1991).[12] With the momentum of Windows 3.0's commercial success, the release of Windows 95 was framed as a giant leap forward – and accompanied by a high-profile marketing campaign. The entire operating system was overhauled, with a new user

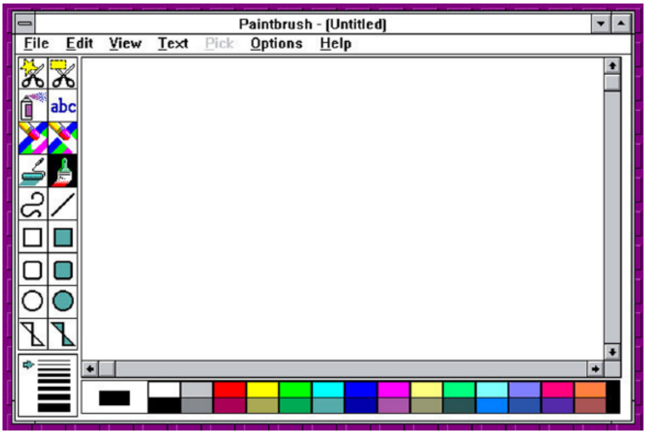
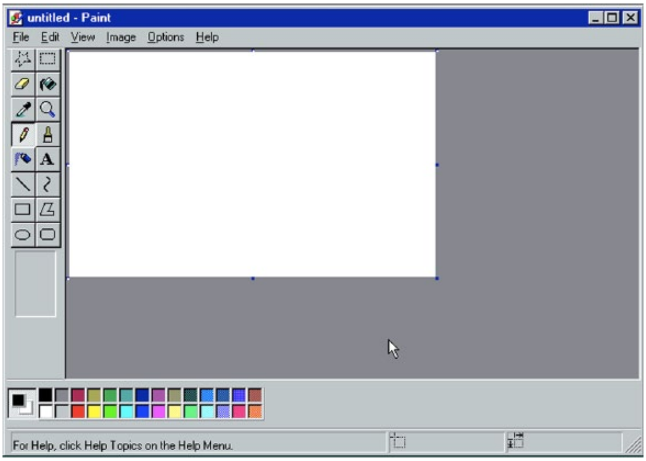


Figure 3 The interface of Microsoft Paintbrush running on Windows 3.1. Reproduced with permission of Microsoft.

interface (this was the origin of the 'Start' menu) and improved graphics handling. Microsoft Paintbrush was refreshed as well, although the 'updates' were largely superficial: the 'pencil' tool returned, and the 'brush' tool had a slightly different set of controls. Probably most significant to this history, the name was truncated to Microsoft Paint (lending itself to the further cultural abbreviation of 'MS Paint'). In terms of interface and functionality, Microsoft Paint remained familiar to its predecessor (see Figure 4). While embedded applications, such as Windows Paint and Windows Write, had been motivators for selling earlier versions of Windows as an after-market addition to PCs already running DOS, Microsoft had changed the landscape of personal computing by 1995. New licensing deals with IBM (as well as PC 'clone' manufacturers) arranged for Windows 95 to come pre-installed on IBM and IBM-compatible machines. Paint software was no longer needed to help entice skeptical potential customers to purchase an untested operating system, thus terminating the economic motivation for keeping the program up to date (Tandy, 2014, personal communication).

MS Paint did not receive another significant update until the commercial release of Windows Vista in 2007.[13] Thus, the graphics program that was most available during more than a decade of intensifying internet usage and meme production, the period from 1995–2007, was one inherited directly from the painting methods and tools of the 1980s. What the history of this program shows is that this was the result of a string of historical contingencies; each 'version' of MS Paint produced between 1985 and 1995 was subject to less than optimal conditions. Curtailed development, tight deadlines, intentionally limited designs, and waning economic motivations all interceded at various points to keep the Windows standard graphics program relatively under-developed. As I will show next, the formal properties of MS Paint that resulted from this under-development produced



For Help, click Help Topics on the Help Menu.

Figure 4 The interface of Microsoft Paint running in Windows 95. Reproduced with permission of Microsoft.

a particular visual character that would go on to influence the style of many memes from the last decade.

MS Paint Style: Mouse-Drawn Bitmaps

In their discussion of ‘glitch’, Olga Goriunova and Alexei Shulgin (2008: 113) write:

> There are moments in the history of computer technology that are rich in computer functionality producing distinct aesthetics. At such times, computer functionality reveals itself through technological limitations. Bottlenecks, such as processor speed, screen resolution, color depth, or network bandwidth – 4-bit, 8-bit music, 16-color pixelized visuals, slow rendering, compressed image and video with artifacts – create an authentic computer aesthetics, that is, the aesthetics of low-tech today.

Among the programs used to produce memes, none is more associated with a specific ‘aesthetic of low-tech’ than MS Paint.[14] As anyone who has used the program can attest, it takes a great deal of craft and skill to produce images with MS Paint that ‘look good’ (or as Nick Douglas might say, as per his article in this issue, ‘don’t look shitty’). Images produced in MS Paint are often immediately identifiable by this ‘shittiness’ – erratic mouse-drawn lines, uneven edges, blocky dumps of color.[15] As Goriunova and Shulgin imply, this visual character is not arbitrary. It is shaped by the reality of MS Paint *as software*. MS Paint, even in its post-1995 incarnations, is the inheritor of decisions made and techniques adopted during much earlier periods. Of these prior developments, two shaped the visual character of

MS Paint more than any other: the choice to develop MS Paint as a bitmap-based program, and its close association with the mouse.[16] In what follows, I will provide further context for this particular combination of software and hardware, and read their visual impact in a contemporary image of a popular meme – 'Rage Guy' (see Figure 5).[17] Although MS Paint leaves no digital fingerprint as it were, it is widely held that this Rage Comic, anonymously posted to 4chan in 2008, was drawn in MS Paint (Know Your Meme, 2014c).[18]

So what can we see in this iconic image? The image is sparse and dominated by a default white background. The exact repetition of the seated figure implies the use of 'copy' and 'paste'. There are only five colors in the image: single values for white, black, blue, brown, and red.[19] Except for the heavy lines dividing the four panels and the red text, all of the lines are four pixels wide. Details such as the top corner of the toilet tank, the outstretched fingers, or right pupil of the face all capture moments of transition between perfectly straight lines and shakily curved ones. The ends of various strokes indicate the brush used was likely a circle, while sharp corners where erasure is visible along the figure's butt in panel two and on the top right edge of the face in panel four, indicate the eraser tool was likely square. Unsurprisingly, a round brush and square eraser are the default on the Windows 95 version of MS Paint. The overall character of this image,

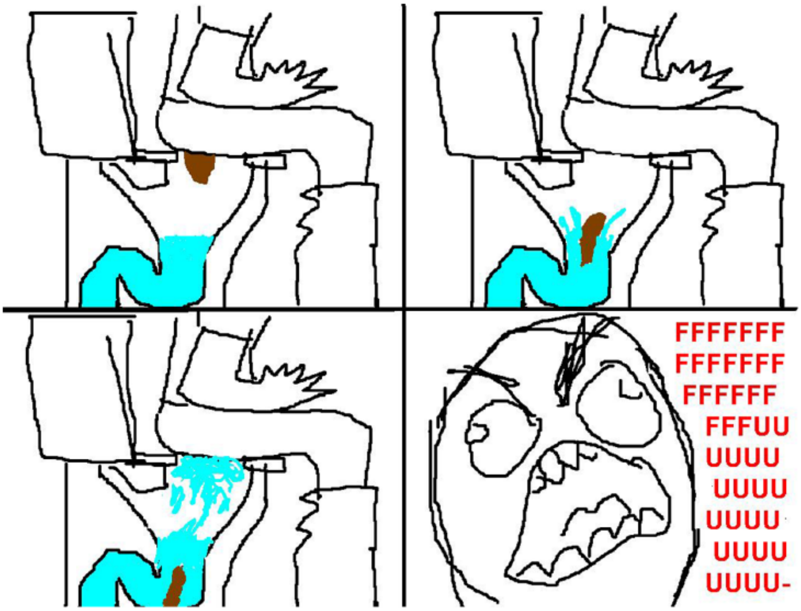


Figure 5 According to Know Your Meme and other diverse and unverified sources online, this image was the first to depict the now-iconic 'Rage Face' and spawn the subsequent Rage Comic genre. Image courtesy of knowyourmeme.com

however, is influenced by more than just brush size and shape. In order to understand the visual character of MS Paint on a deeper level, we need to account for its relationship to the bitmap technique of graphics, and the popularity of the mouse as an input device.

The bitmap

The portions of the Rage Comic that transition between perfectly straight horizontals or verticals and rough, irregular curves make it easy to see the jagged line quality of images produced with MS Paint. In the parlance of computer graphics, these stair-step edges to curves are called 'jaggies' and they are one consequence of representing images as bitmaps rather than geometric objects. A technique called 'anti-aliasing' was developed to eliminate them as early as 1973 (Shoup, 2001; Van Dam, 1984). Anti-aliasing is an algorithmic process for placing pixels of intermediary value at the borders between larger areas of color. Zooming in on the original Rage Comic alongside a more recent, anti-aliased version makes this apparent (see Figure 6): different shades of gray have been automatically placed along the edge of curves and corners. These intermediate tones soften the lines and make the individual pixels of the image less visible when viewed at normal size. The limitations of personal computer hardware in 1985 made it technically impractical for the original Windows Paint to use this technique. But even more significantly, in 1995 when the hardware of the average personal computer was technically capable of anti-aliasing, the tools of the new MS Paint still did not implement this technique.

The two technical factors that determine a system's capacity for anti-aliasing are resolution and color.[20] Resolution is a measure of how many individual pixels make up a single image, while the quality of a bitmap's color is determined by 'bit-depth' or how much information is allotted to each pixel. 'Pixel' is a difficult word to use precisely. Graham Harwood (2008: 215) calls it 'the basic unit of programmable color in our seeing machines', but this basic unit has different meanings for hardware and software. For hardware,

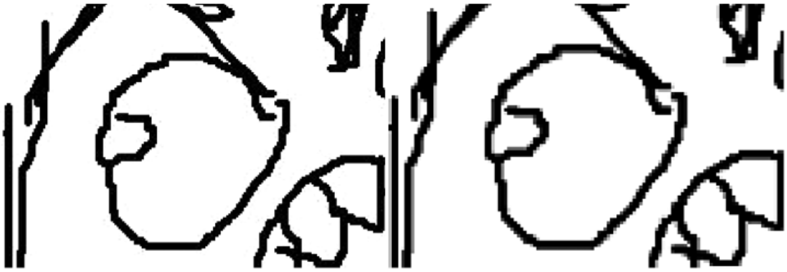


Figure 6 The image on the left is a detail of the Rage Comic that shows the blocky quality of the lines. The image on the right shows the same lines altered with an anti-aliasing algorithm that adds intermediary values and softens the edges.

like cathode ray or LCD screens, the word ‘pixel’ or ‘picture element’ (Lyon, 2006) refers to the smallest physical component that can display a full range of values. For software, the term refers to the numerical storage of value for the smallest point of color. The resolution of an image as it appears to the human eye is therefore determined by the number and density of *both* types of pixels. A high resolution image file requires a high resolution display to render it as such. ‘Bit-depth’ is similarly dependent on the performance of both hardware and software. Bit-depth refers to the amount of information (literally the number of bits) that a graphics system can assign to each pixel. Alvy Smith (2001: 6), who worked with Dick Shoup on SuperPaint and later developed his own paint systems, summarizes the evolution of bit-depth:

> 1-bit, 3-bit, 8-bit, 24-bit, and 32-bit. These correspond, respectively, to 2 colors (black and white), 8 colors, 256 colors, 16.7 million colors, and 16.7 million colors plus 256 levels of transparency. I give the 1-bit and 3-bit systems short shrift. The 8-bit, or 256-color, systems made digital painting a real tool in video. The 24-bit and 32-bit paint systems are required for film use.

While anti-aliasing effectively ‘solves’ the problem of jaggies in bitmaps, it requires a system of sufficiently high resolution and bit-depth. And even when an individual machine meets these criteria, the software must still be programmed to take advantage of it. In 1990, Windows 3.0 was programmed to accommodate 24-bit or ‘true’ color (16.7 million values) for the first time (Bellin and Del Frate, 1990).[21] In 1995, five years after this requisite technical advance, the tools of MS Paint were not updated to take advantage of any visually smoothing algorithm. The brush tool, circle tool, bezier line tool – none of these have any anti-aliasing. Thus, lines drawn with MS Paint in 1995 and after automatically contained a visible trace of the hardware limitations of the 1980s.[22]

The mouse

‘By far the most popular [drawing device] is the mouse’, wrote Jeffrey Young (1984: 127). Yet he went on to clarify:

> In the computer industry, [1984] has been the ‘Year of the Mouse.’ The mouse has become synonymous with user-friendly personal computers … But painting with a mouse is not as easy as some people would have you believe. After years of grasping a pencil, pen or brush between index finger and thumb, holding the small rectangular mouse has an entirely alien feel (p. 129)

This ‘alien feel’ has very real consequences for the visual quality of the images produced with a mouse. Producing lines with a shoulder–elbow–wrist movement, especially when paired with a lack of anti-aliasing and a set of tools modulated by pixel widths, creates a recognizable visual character.[23] The two images in Figure 7, one (likely) drawn with a mouse and the other with a tablet, have noticeably different line qualities, but this difference is not something to which we can attach a number in the way we can distinguish between 8-bit and 24-bit color. One can see in long curves such as those in the Rage Comic's eye or mouth, where the generally convex shape wavers into momentary portions of concavity.[24] And yet, in spite of this, Windows Paint was specifically designed and programmed to be used with a mouse.

PC Paintbrush (the precursor to 1990's Microsoft Paintbrush) made the relationship between mice and 'paint' programs explicit in the years between 1985 and 1990. In spite of the fact that mice had become the *industry* standard for use with graphical user interfaces (GUIs), there was still widespread *consumer* resistance to them throughout much of the 1980s.[25] 'Although the mouse itself is a small part of the expense', wrote Phil Lemmons (1983: 48), 'it is a symbol of this approach to software, and some computer users have been heard to mutter, "What price mice?"' ZSoft and Microsoft alike determined that PC Paintbrush was a useful means for getting people to purchase and use mice, and so for several years in the late 1980s, PC Paintbrush was sold bundled together with a Microsoft-brand mouse (Zachmann, 2014, personal communication). This bundling communicated that bitmap painting programs like PC Paintbrush (and later Microsoft Paintbrush) were compelling reasons to buy and use a mouse. Once the mouse became the standard input device sold with all personal computers (a trend that was complete by 1995) this relationship between hardware and software no longer needed endorsement; it was simply the default.

That the combination of bitmap and mouse have had such visible effects on the visual character of images produced with MS Paint reinforces that MS Paint was never about painting. MS Paint was not meant as a remediation of moving

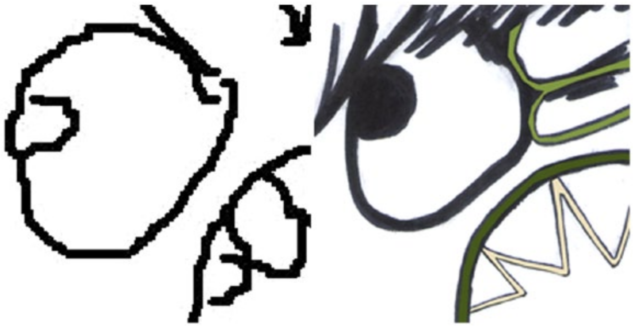


Figure 7 The image on the left is a detail of the 'original' Rage Comic. On the right is the detail of a Rage Comic parody of a cartoon character, drawn using a tablet and uploaded to deviantart by user GlandEnce.

pigment on canvas with a brush, and therefore did not need to consider an input device with fidelity to the art historical praxis of painting. Instead, MS Paint was designed as a mediation of two separate technologies: the bitmap and the mouse. The visual dimensions of resolution and bit-depth and the physical experience of using a mouse are therefore the hallmarks of 'an authentic computer aesthetics'; they produce the visible traces of techniques and devices unique to digital computers. We can see evidence of this in the use of MS Paint after 1995. The use of MS Paint is not associated with painting forums or communities of fine art lovers. It is the graphics tool of internet memes and computer culture – of communities whose identities are tied to the deliberate consumption of specifically digital media. In the next section, I will examine two contemporary programs, each inspired by MS Paint's past, and each with a drastically different relationship to visible traces of digital technologies.

Mediation and Remediation after MS Paint

The use of MS Paint was (and is) arguably a meme in itself. Starting in 2000, there were numerous threads on the comedy forums of website Something Awful with labels like 'MSPaint your favourite film!' or 'I Will Draw the Pornography of Your Choice … in MS Paint' (Somethingawful.com, 2001). In these instances, the use of MS Paint was adopted specifically because it imbued images with the characteristic low-tech quality discussed above. We've already examined Rage Comics, but there is no shortage of other genres and characters (memes like fsjal, or Dolan Duck) that specifically embraced this visual character even if potentially produced with software other than MS Paint (see Figure 8). The quality of lines in images of fsjal indicate that many were likely made using a vector-based program, but their blocky figures on blank backgrounds and limited color palettes match the quality of many images produced with MS Paint. Images of Dolan Duck adhere even closer to this visual character. Though, instead of original mouse-drawn lines, many of the Dolan Duck images involve the cutting and pasting of a common set of characters with new textual additions (not to mention the 4-panel frame common to Rage Comics). But, MS Paint's influence goes beyond just images and memes. It has also influenced the design and use of new pieces of software. To understand what these new programs might 'inherit' from MS Paint requires a further consideration of remediation.

Remediation is a popular concept in the discussion of new media, from Marshall McLuhan's (1964) influential claim that 'the content of a medium is always another medium', to Jay David Bolter and David Grusin's 2000 book *Remediation: Understanding New Media*, to Lev Manovich's (2013) more recent discussion of the computer as a 'meta-medium'. All of these contribute to a consideration of how one medium simulates or *re*mediates another. It is impossible to fully understand MS Paint without a concept of remediation. The iconography used to represent the tools, the functioning of the tools themselves – both of these rely on a knowledge and concept of the media technologies they purport to reference, simulate, and remediate.

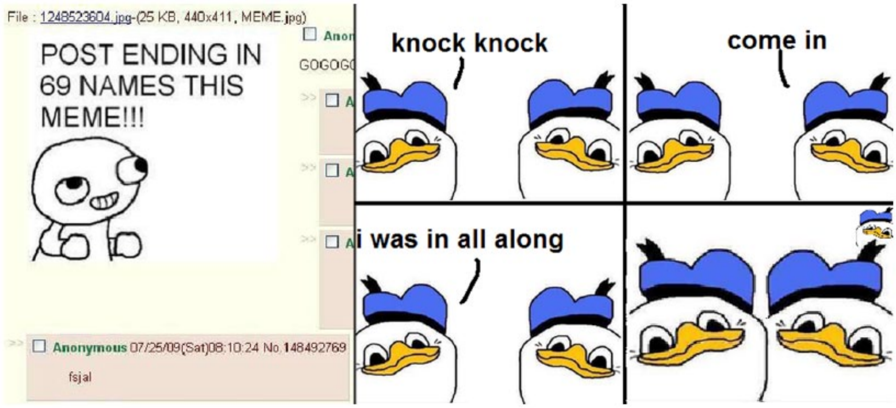
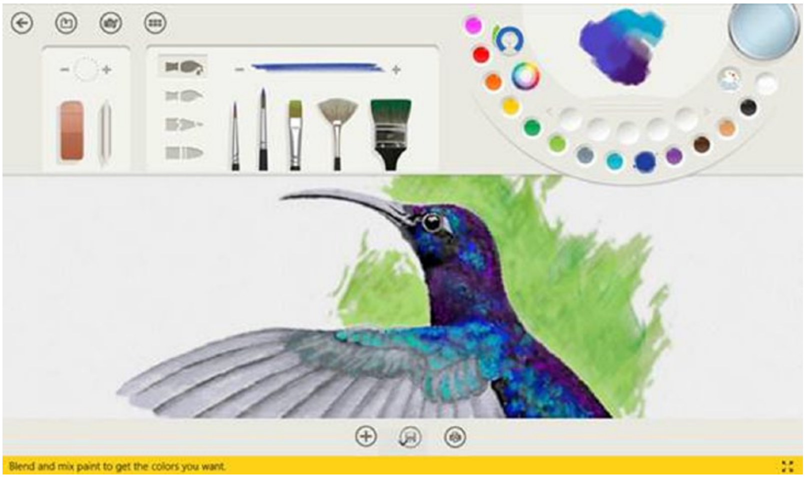

File : 1248523604.jpg-(25 KB, 440x411, MEME.jpg)
POST ENDING IN 69 NAMES THIS MEME!!!
Anonymous 07/25/09(Sat)08:10:24 No.148492769
fsjal
knock knock
come in
i was in all along

Figure 8 Two other examples of an MS Paint-style in images made after 2008: fsjal, and Dolan Duck (Know Your Meme, 2014a, 2014b). Images courtesy of knowyourmeme.com

But if knowledge of remediation is necessary, it is not sufficient. There are limits to what the concept of remediation can reveal about a particular piece of software. Focusing on the formal features of MS Paint that ‘fail’ in their imitation of traditional painting allows us to shift our focus from remediation to mediation. Analyzing MS Paint’s ‘authentic digital aesthetics’ is valuable because it enables a consideration of digital media as an autonomous sphere of production and value. To illustrate, I discuss two recently produced graphics programs, both inspired by MS Paint, each with a radically different relationship to digital remediation: the free, touchscreen-enabled Microsoft app Fresh Paint, and the Rage Maker in-browser software program used by the Rage Comic-devoted f7u12 subreddit.[26]

Fresh Paint

Beginning in 2012, alongside the release of Windows 8, there was a rash of publicity directed at a new Microsoft ‘app’ named Fresh Paint (see Figure 9) (Warren, 2013). This new, ‘fresh’, alternative to MS Paint provides high definition renderings of a white plastic palette, different daubs of hue, and even a pool of water that ripples and changes color as you touch it to ‘clean’ your ‘brush’. The program offers an array of paper textures and a ‘fan’ button to dry the otherwise quick-to-blend-and-smear colors (Microsoft, 2012). This emphasis on the simulated remediation of traditional painting, combined with the increase in screen resolution and graphics performance of contemporary computers, means that visual traces of the digital (like those discussed above) do not appear in images made with Fresh Paint. These images are still based on a version of the bitmap technique of image production, but realized in such a drastically enhanced technical context that the illusion of non-digital image making is more or less complete. Fresh Paint exists on the other side of the ‘bottleneck’ that shaped the aesthetics



Blend and mix paint to get the colors you want

Figure 9 The interface of Microsoft's 2012 app: Fresh Paint. Image courtesy of Microsoft.

of MS Paint. MS Paint provided the tools and interface to consider images *as collections of pixels*; in Fresh Paint, those pixels are only there to produce the impression of strokes, daubs, and pools of imaginary pigment on imaginary paper. The program's formal properties actively discourage a consideration of the digital quality of digital images.

From the point of view of Microsoft, however, Fresh Paint serves a remarkably similar role to that of the original Windows Paint. Like Windows Paint in relation to the desktop-based GUI, Fresh Paint is inextricably tied up with a new paradigm of personal computing embodied in the app-centric, tiled interface of Windows 8. Fresh Paint is designed to take specific advantage of a newly mainstreamed input device – the touch screen of tablets and smartphones. It is marketed as a free supplement to Microsoft's operating system, intended to motivate sales. Unlike MS Paint, Fresh Paint is devoted to the concept of remediation – of making 'painting' about painting. With Fresh Paint, Microsoft moves away from the visible digital traces that have historically characterized the use of MS Paint. Therefore, in order to find a new program that handles technical limitation in a mode similar to that of MS Paint, we need to look beyond Microsoft itself.

Rage Maker

Programmed in the years after Rage Comics became popular in 2008, Rage Maker (accessible at the time of writing at ragemaker.net) is a custom Flash-based web-application that allows users to create new Rage Comic images in their browser (see Figure 10). Rage Maker's interface and toolset departs significantly from the drawing/painting paradigm that held sway during the original creation of MS Paint. Rage Maker is a hybrid between object- and bitmap-based techniques. The interface provides an outer 'frame' inside of

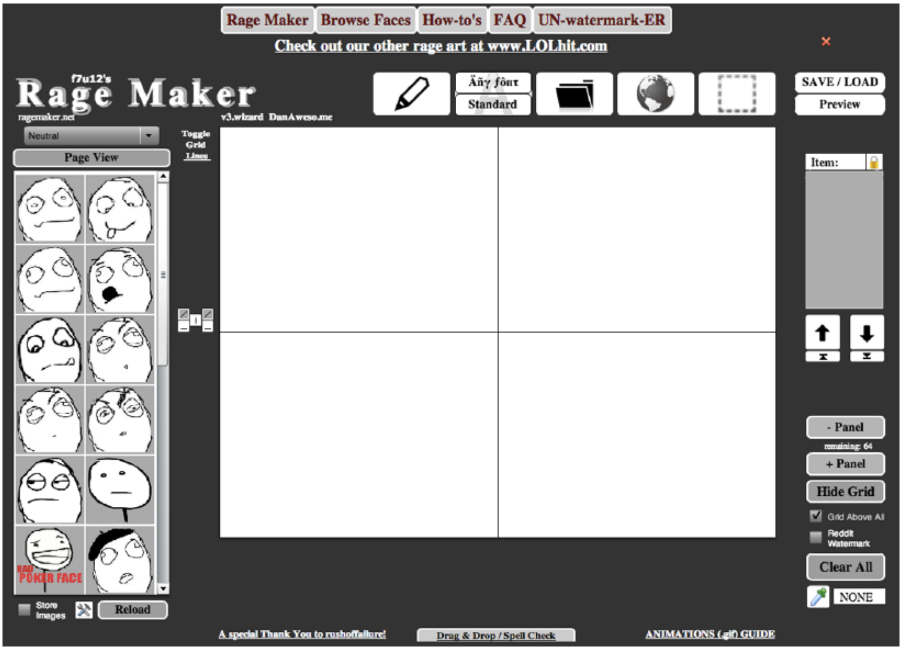


Figure 10 The interface of Rage Maker available at ragemaker.net

which users are able to create different individual 'frame' objects. These interior frames maintain their individual characteristics and can be moved, repositioned, and placed 'in front of' or 'behind' each other like objects in a geometry-based drawing program. Each frame can contain either a pre-drawn image selected from a large repository, a type tool, or a miniature bitmap painting tool with pencil, shapes, and 216 possible colors.[27]

The library of pre-made images and the tools provided by Rage Maker make it clear that the program is meant to produce a genre of images traditionally made and associated with MS Paint. However, Rage Maker is by no means a simulation of MS Paint. It provides many new affordances (image library, layers, grid) and omits others. It even automatically anti-aliases its lines. However, what is important to this discussion is not that Rage Maker imposes a *specific* technical limitation; it is that the software produces images that make technical limitations visible. Rage Maker is not subject to the same anti-aliasing limitations as MS Paint, but it nonetheless records the visual traces of software. It is limited to 216 'web safe' colors. It has a fixed width of 651 pixels and a resolution of 72 pixels/inch. While images made in Photoshop can aspire to the aesthetics of photo-realism, and images made in Fresh Paint can aspire to the aesthetics of paper and pigment, images made in Rage Maker – like those made in MS Paint – cannot aspire to any aesthetics other than those of a mouse-drawn bitmap: an 'authentic computer aesthetics'.

The differences between these two programs – Fresh Paint, Rage Maker –illustrate the varied cultural impact MS Paint has had. A lapsed commercial product, a source of irony and irreverence, the bargain basement of graphics software, the foundation of online artistic expression – MS Paint has been (and still is) many things to many people. And although a program like Rage Maker might seem 'closer' or more similar to MS Paint, given the community it serves and the type of images it produces, the longer history of Microsoft Paintbrush and Windows Paint reminds us that we cannot predict what role Fresh Paint might play a decade from now, or what emergent communities of practice it might someday support after it too has fallen from mainstream attention.

Conclusion

The history of MS Paint shows that a diverse set of historical contingencies resulted in the technical limitations of the version of MS Paint released in 1995. These technical limitations had a specific impact on the visual character of images produced with the program for the next decade and beyond. The effect of this can be read in the other graphics programs of recent years: Fresh Paint imitating MS Paint's original economic role, and Rage Maker paralleling its relationship to technical limitation. Taken together, these three approaches to MS Paint provide a counterfactual to the dominant ideology of technological progress. The popular use and cultural influence of MS Paint were not caused by technical advancements. Instead, it was MS Paint's relative underdevelopment that opened up space and opportunity for a style of production that would not have been what it was had its developers scrambled to keep the program up to date. While more rigorously updated programs like Photoshop have become industry tools, subjects of technical manuals, and objects of academic case studies of digital media (Manovich. 2001, 2011), MS Paint has escaped serious notice during its three decades of existence. While concepts of technological innovation, or remediation, are valuable tools for thinking through the cultural role of computing, we should remember that they are not the only tools, nor always the most apt.

Of course, it can be difficult to pay attention to things that tend to escape attention, but this is all the more reason why we should make the effort. 'Computers are most powerful when least noticed', writes Geoffrey Winthrop-Young (2010: 186), 'and they are least noticed when most empowering.' When and how has MS Paint been noticed, and who and what does it empower? Photoshop and programs like it are included in lists and articles trumpeting 'a software canon', 'the most influential software programs ever', or 'best tech products of all time' (Ford, 2014; Kirschenbaum, 2013; Null, 2007). No version of MS Paint has ever made it to such a list; it has simply soldiered along. In the end, then, this article is not just about a particular program or set of technologies or visual style, but also a provocation about these superlative software lists MS

Paint does not qualify for. To which unwritten list *does* MS Paint belong, and which other consistently unconsidered and furiously influential programs belong there beside it?

Acknowledgements

This article would not exist save for Laine Nooney's enthusiasm and aptitude for software-focused research, and Jacob Gaboury's considerable knowledge and generosity. I am also indebted to Amy Stevenson and Roy Zimmermann of Microsoft, as well to Mark Zachmann, Tandy Trower, Jody Snodgrass, and Dan McCabe, all of whom shared first-hand accounts of MS Paint's history. Additional editorial feedback was provided by Laura Portwood-Stacer and Whitney Phillips. Finally, my endless thanks to Michelle Forelle for keeping me honest, improving my prose, and supporting this work.

Notes

1. Windows Vista incorporated the 'ribbon interface' into MS Paint. This did not alter many of the functions of MS Paint but did make a pretty significant change to the interface; although, by 2007, the association of MS Paint with a particular visual style was already well underway.
2. From the 1986 Adobe annual report:

   > To date most illustration programs intended for the personal computer market fall into one of two categories. There are the "paint" style programs, and there are the "object" oriented programs. "Paint" style programs dictate that the user construct a fixed, resolution dependent "bit map" image through sketching with a mouse. These images have a very characteristic appearance. They are rough images made up of little black squares that produce jagged lines, tints, and patterns. These programs are unsatisfactory for serious illustration work because of the primitive, stylized look of the illustrations that result, and the large size of the files they produce. "Object" oriented programs solve the problem of jagged appearance and large file size, but are typically difficult to use. They restrict the kinds of illustrations that can be created because they are limited to produce only what the underlying computer graphics system will allow. (Adobe, 1986: 8)

3. In order for bitmaps to work with the monitors of the time, Shoup not only had to write custom software but also assemble a new piece of hardware called a frame buffer that could communicate in the necessary two dimensions (Shoup, 2001; Smith, 2001).
4. Alvy Ray Smith (2001) mentions the work of Joan Miller of Bell Labs of a paint program using a 3-bit frame buffer, three years before Shoup's work, but does not go into any detail about this earlier precursor. Research indicates Miller's work has been largely undocumented.
5. David Liddle (1998) discusses the explicit marketing of the Xerox Star in his presentation during the Star's final public demonstration at Xerox PARC.
6. This divide was reified not by Microsoft, but by their competitor, Apple, in the two years before Windows' release. Apple released the Lisa computer in 1983, costing $10,000 and catering to 'the needs of the business person'. In 1984, they released the Macintosh, which cost $2,500, and was designed specifically to be 'affordable to individuals rather than corporations'. And which graphics

programs did these two systems come with? The Lisa included (geometry-based) LisaDraw, while the Macintosh came with (bitmap-based) MacPaint.

7. There was a demonstration of Windows given at the Comdex convention in 1983 that appears to feature a ‘Paint’ application depicting a bitmap drawing of a tree. This program was in fact a mock-up, based on the code for Windows Chart rather than a fully functional version of Windows Paint (McCabe, 2014, personal communication).
8. The Apple Macintosh advertisements of the time consistently featured a Macintosh running MacPaint with some slogan drawn in script handwriting as the central visual element.
9. McCabe (2014, personal communication) characterizes the development of Windows Paint as ‘standing on the shoulders of giants’ in that it drew inspiration from the techniques developed by several other previous programs.
10. The ability to run PC Paintbrush in 16 colors was dependent on the purchase and installation of an after-market graphics card. In fact, the original version of PC Paintbrush was commissioned to motivate the sale of such graphics cards (Zachmann, 2014, personal communication).
11. Microsoft Paintbrush was also the first MS Paint to program support for a two-button mouse. The color selection palette is split between a foreground and background that were selected by the left and right mouse buttons, respectively.
12. According to the sources cited above, the number of copies sold of Windows 1.0 in the first year would have been significantly less than 500,000, while the number of Windows 95 sold in its first year would be estimated at 40 million. While these numbers are largely imprecise, they nonetheless sketch the extreme difference in commercial performance during these time periods.
13. The update to Windows XP in 2001 would introduce an updated graphical device interface (GDI) that allowed MS Paint to natively produce .jpg and .gif image files. This had a large effect on MS Paint’s usefulness for online image production, but did not alter the interface or tools in anyway.
14. Special mention should be made of Gary’s Mod – a modification of the popular Team Fortress computer game that has been widely used in the production of glitchy, low-tech appearing videos and images. However, Gary’s Mod requires a significantly higher level of technical knowledge to use compared to MS Paint.
15. Nick Douglas goes to much greater lengths to explore the various branches of this aesthetic, that he terms ‘Internet Ugly’ in his article in this issue.
16. Among the formal features of MS Paint, the lack of layers, the inconsistent availability of compressed image formats, and the integration with the ‘Print Screen’ button on Windows machines have all had significant effects on how the program has been used to produce images, but these properties are less likely to leave discernible visible traces in the images themselves.
17. Users of popular link and image sharing site reddit created a ‘subreddit’ or themed discussion area called ‘f7u12’ specifically for the sharing of Rage Comics in 2009. It can be visited online at: http://www.reddit.com/r/fffffffuuuuuuuuuuuu/.
18. The historical accuracy of this claim has not been and likely cannot be definitively verified. The majority of sources that claim this image was the first Rage Comic (Know Your Meme included) cite the website Encyclopedia Dramatica as the source of this claim (and the archived

version of the image). Encyclopedia Dramatica is a community-maintained documentation of meme culture maintained using wiki software that is specifically constructed as an offensive and aggressive site of information and disinformation. Therefore, while it is likely that the above image was one of the earliest Rage Comics, was made in MS Paint, and was posted to 4chan in 2008, none of these points can be definitively authenticated.

19. Careful readers might detect small variations of color in the fields of brown and blue. These are the digital artifacts of the .jpg compression process, and were not values placed by the mouse-directed tools of MS Paint.
20. Once again, a quote from Jeffrey Young (1984):

> [The bit-map approach] is a software programming sleight of hand where the screen is divided into thousands of dots, or pixels, corresponding to the intersections of the vertical and horizontal scan lines that comprise a cathode ray screen. Each of these dots can be controlled by the paint programming—the more dots, the higher the resolution of the image … But where resolution is essential in determining the on-screen appearance of a drawing, so is color. (p. 129)

21. Any individual computer lacking a 24-bit-capable card would instead rely on an algorithmic accommodation called ‘dithering’. If a piece of software (such as the 24-bit-capable Microsoft Paintbrush) contained a color value outside the bit-depth of the frame buffer, that value could be approximated by a small grid of two similar colors. If the system didn’t have enough bits to store the correct value for ‘purple’ it could compensate with a checkerboard of ‘red’ and ‘blue’.
22. The version of MS Paint that shipped with Windows 7 in 2009 added anti-aliasing to its brush tool. This feature could be disabled.
23. From a sidebar in the same *Personal Computing* article, artist Lauretta Jones (cited in Young, 1984: 130) explains: ‘When I’m using a mouse my fingers are useless. I end up drawing with my arm, shoulder, and even back muscles. And that part of me doesn’t know how to draw.’
24. Bitmap painting programs do not *require* the use of a mouse. In 1973, Shoup’s influential SuperPaint program used a Summagraphics tablet for input – ‘a flat, markless surface’, that ‘opaquely concealed a grid of wires … that sequentially transmitted electrical pulses’ (Nappi, 2013:164) and allowed for the use of a pencil-like stylus.
25. Following Douglas Engelbart’s 1968 demonstration of the NLS (oN-Line System) – the first computer system to use a mouse – the input device became associated with the GUI-based paradigm of computing. In 1975, Harold Hall organized a team at Xerox to develop this paradigm into a new business product (Liddle, 1998). The result was the mouse-dependent Xerox Star, released to the public in 1981; over the next four years, Apple’s Lisa and Macintosh, and the Windows OS would all implement mouse-driven input.
26. A subreddit is a portion of reddit’s larger platform devoted to a single topic or type of content. These subreddits can be started and named by any user on any topic, therefore the number and variety of such subreddits is considerable.

27. The seemingly arbitrary number of 216 colors is the result of the defining of a set of ‘web safe’ colors. In much the same way MS Paint has been programmed relative to the technical limitations of its own time, Rage Maker is shaped by the current limitations placed on web applications.

References

Adobe Systems Incorporated (1986) *Adobe Annual Report*. Palo Alto, CA. Available at: http://corphist.computerhistory.org/corphist/documents/doc-447f52ae8241b.pdf (accessed 30 May 2014).

Bellin A and Del Frate P (1990) True Color for Windows. *BYTE*, 1 December: 281–288.

Bolter J and Grusin R (2000) *Remediation: Understanding New Media*. Cambridge, MA: MIT Press.

Burns C (1996) A year later, WinNT still in Win 95’s Way. *Network World*, 26 August: 25.

Coreldraw (2014) Graphic design software – CorelDRAW Graphics Suite X7. Available at: http://www.coreldraw.com/us/product/graphic-design-software/?pid=prod5190074 (accessed 30 May 2014).

Eglash R (2008) Computing power. In: Fuller M (ed.) *Software Studies: A Lexicon*. Cambridge, MA: MIT Press, 55–63.

Ford P (2014) The Great Works of Software. *Medium*, 30 April. Available at: https://medium.com/message/705b87339971 (accessed 30 May 2014).

Fox S and Rainie L (2014) The Web at 25 in the US. *Pew Research Center’s Internet & American Life Project*, 27 February. Available at: http://www.pewinternet.org/2014/02/27/part-1-how-the-internet-has-woven-itself-into-american-life/ (accessed 30 May 2014).

Godwin M (1994) Meme, counter-meme. *Wired*, 1 October. Available at: http://archive.wired.com/wired/archive/2.10/godwin.if_pr.html (accessed 25 May 2014).

Goriunova O and Shulgin A (2008) Glitch. In: Fuller M (ed.) *Software Studies: A Lexicon*. Cambridge, MA: MIT Press, 110–118.

Harwood G (2008) Pixel. In: Fuller M (ed.) *Software Studies: A Lexicon*. Cambridge, MA: MIT Press, 213–215.

Huhtamo E and Parikka J (2011) *Media Archaeology*. Berkeley: University of California Press.

Kirschenbaum M (2013) The 10 most influential software programs of all time. *Brow Beat*, 30 July. Available at: http://www.slate.com/blogs/browbeat/2013/07/30/_10_most_influential_software_programs_of_all_time_from_sabre_to_minecraft.html (accessed 30 May 2014).

Know Your Meme (2014a) Dolan. Available at: http://knowyourmeme.com/memes/dolan (accessed 30 May 2014).

Know Your Meme (2014b) fsjal. Available at: http://knowyourmeme.com/memes/fsjal (accessed 30 May 2014).

Know Your Meme (2014c) Rage Guy (FFFFFUUUUUUUU-). Available at: http://knowyourmeme.com/memes/rage-guy-fffffuuuuuuuu (accessed 30 May 2014).

Lemmons P (1983) Microsoft Windows: A mouse with modest requirements. *BYTE*, 1 December: 48–54.

Liddle D (1998) *What Were We Doing Anyway? How The Star and XNS Came To Be*. Palo Alto, CA, 17 June. Available at: https://www.youtube.com/watch?v=puyhwV4aqJs (accessed 30 May 2014).

Lindsay C (2003) From the shadows: Users as designers, producers, marketers, distributors, and technical support. In: Oudshoorn N, Pinch T (eds) *How Users Matter: The Co-Construction of Users and Technology*. Cambridge, MA: MIT Press, 29–50.

Livingston B (1991) Cooking up a Windows networking strategy. *Infoworld*, 21 October: S1.

Lyon R (2006) A brief history of 'pixel'. In: *IS&T/SPIE Symposium on Electronic Imaging*, San Jose, CA, 15–19 January.

Manovich L (2001) *The Language of New Media*. Cambridge, MA: MIT Press.

Manovich L (2011) Inside Photoshop. *Computational Culture* 1(1). Available at: http://computationalculture.net/ (accessed 10 March 2014).

Manovich L (2013) *Software Takes Command*. New York: Bloomsbury.

McLuhan M (1964) *Understanding Media: The Extensions of Man*. New York: McGraw-Hill.

Nakamura L (2002) *Cybertypes*. New York: Routledge.

Nappi M (2013) Drawing w/Digits_Painting w/Pixels: Selected artworks of the gesture over 50 years. *Leonardo* 46(2): 63–169.

Null C (2007) The 50 best tech products of all time. *PCWorld*, 2 April. Available at: http://www.pcworld.com/article/130207/article.html (accessed 30 May 2014).

Microsoft (2012) Fresh Paint app for Windows in the Windows store. Available at: http://apps.microsoft.com/windows/en-us/app/fresh-paint/1926e0a0-5e41-48e1-ba68-be35f2266a03 (accessed 30 May 2014).

Shoup R (2001) SuperPaint: An early frame buffer graphics system. *IEEE Annals of the History of Computing* 23(2): 32–37.

Smith A (2001) Digital Paint systems: An anecdotal and historical overview. *IEEE Annals of the History of Computing* 23(2): 4–30.

Sutherland I (1963) Sketchpad: A man–machine graphical communication system. PhD thesis, Massachusetts Institute of Technology.

Tedre M and Eglash R (2008) Ethnocomputing. In: Fuller M (ed.) *Software Studies: A Lexicon*. Cambridge, MA: MIT Press, 92–100.

Van Dam A (1984) Computer software for graphics. *Scientific American*, 1 September: 146–159.

Warren T (2013) Microsoft's Fresh Paint drawing app overhauled for Windows 8.1 and Windows Phone. *The Verge*, 11 October. Available at: http://www.theverge.com/2013/10/11/4827030/fresh-paint-windows-8-1-windows-phone-updates (accessed 30 May 2014).

Watt P (1987) Few doing Windows. *Computerworld*, 2 March: 1, 14.

Winthrop-Young G (2010) Hardware/software/wetware. In: Mitchell W, Hansen M (eds) *Critical Terms For Media Studies*. Chicago: University of Chicago Press, 186–198.

Young J (1984) How to 'paint' with your computer. *Personal Computing*, 1 November: 126–137.

Patrick Davison is pursuing his doctoral degree in Media Studies from NYU's department of Media, Culture, and Communication. His work focuses on the marginal media of software history and discursive constructions of 'the user'. He lives and writes in Los Angeles.

Email: patrick.davison@gmail.com