New Media Software Tools for Personal Computers:

A History, An Arc, A Lament, Some Prophecy

By James Khazar

Scope of coverage

While there have been many proprietary and customized tools used by artists and designers, I shall largely, though not completely, limit the scope of this paper to the history and direction of interactive media tools in the realm of off-the-shelf software for personal computers. Who can say what garage and basement projects have been created in an ad hoc manner only to be used a few times for unique purposes, or what software was developed internally for consumption only by the employees of specific company? There have also been many hi-end tools available within the commercial sphere, such as Quantel's PaintBox and Harry; and custom software on minicomputers, such as David Em's stint as Artist In Residence at NASA's Jet Propulsion Laboratory in Pasadena from 1977 to 1989. Due to their high cost, artists have only been able to use them in the late-night off-hours when they're not being used for revenue or research generating purposes. It is more useful (and more approachable as a subject) in the scope of this paper to look at tools that anyone could purchase and learn to use on their personal computer.

Since my knowledge and expertise do not extend into the realm of music, I will also not cover tools such as MAX/MSP.

New Media Tools For The Personal Computer

The history of the computer in the modern world is well known, but the history of the personal computer, the kind that many of us have on our desktop or in our backpack today is a bit more obscure. I would like to illuminate that history with some of my own recollections.

I bought my first personal computer in 1980. It was an Apple][Plus and cost a little over a thousand dollars. At that time it was the only small computer with any sort of reasonable ability

to create graphics. The color adaptor card in it was the work of the great electronics genius and Apple co-founder, Steve Wozniak, who had created, with only a few chips and at very low cost, a board which could display eight colors: green, violet, blue, orange, white, black, light gray and dark gray. There were other personal computers around at the time, but the Apple brought computer graphics to the masses.

The Apple][Plus was a landmark in the history of computers, but graphics came along almost as an afterthought. The real money and success for Apple was with what was known as the first "killer app," VisiCalc. Robert X. Cringely, in his book Accidental Empires, put it this way: "VisiCalc was a compelling application – an application so important that it alone justified the computer purchase. Such an application was the last element required to turn the microcomputer from a hobbyist's toy into a business machine…The Apple II was a VisiCalc machine"[1]. In 1979 Dan Briklin and Bob Frankston released the first commercial version on the Apple][. As a direct result Apple saw 78% growth in 1980 and 210% growth in 1981. Until Lotus Corporation bought VisiCalc and subsequently stopped production of it in 1985, VisCalc solidified Apples position in the marketplace and saved it from the obscurity that eventually fell to most other personal computers of the era.

On August 12, 1981, IBM, who had up until that time seen the personal computer market as no more than a fad, released its IBM PC, which rapidly started taking marketshare from Apple. By 1982 Apple's 279% growth was nearly matched by IBM's 240%, but by the next year IBM beat the pants off Apple with respective growths of 1300% and 420%. IBM discouraged the use of its PC as a "toy" and charged and extra $30 for color graphics. What had been a great run for a graphics oriented machine was crushed with great deliberation by "Big Blue."

Apple did not stop there however, although it had a few false starts. In 1981 Apple released the Apple III, which was targeted at business. But it wholly failed, selling only 65,000 units in its product's lifetime. Another attempt was made with the Apple Lisa in 1983. It had the first Graphical User Interfaces (GUI) on a personal computer, and had the first mouse as well. But it wholly failed too, with a price well above the average PC at $10,000. It wasn't until 1984, when

Apple introduced the first Macintosh, that graphics became an important part of the personal computing world again.

Meanwhile, competition in the marketplace and a customer thirst for game software as well as business software impelled non-Apple, DOS-based[2] computers to add more graphics capability. IBM's original $30 markup for graphics quickly became part of the standard package with their CGA (Computer Graphics Adapter) card. It was capable of displaying three colors at a time from a palette of six with a resolution of 320 by 240 picture elements, or "pixels."[3] In 1984 IBM introduced an improved standard called EGA (Enhanced Graphics Adapter), which bumped up the resolution to 640 by 350 and the colors to 16 from a palette of 64. Then in 1987 they added the VGA (Video Graphics Array) with 640 by 480 resolution but no more colors. Competitors saw an opportunity and extended the VGA standard to 256 colors, creating SVGA (Super Video Graphics Array), so that by the early 1990's 640 by 480 with 256 colors became the de facto display standard for personal computers.

There was an alphabet soup of display standards through the 1990's, including such initial sets as XGA, MCGA, UXGA, WUXGA, WXGA, WXGA+. Suffice it to say that by the end of the 1990's most standards and adapter cards for displaying graphics had progressed to 1024 by 768 pixels capable of displaying millions of colors[4]. Higher resolutions are becoming more common. The top of the line for personal computers is in Apple's 30-inch Cinema Display with 2560 by 1600 pixels with millions of colors.

There is an obvious trend, in these aspects, of personal computer's ability to display at higher and higher resolutions, on larger and larger screens, for less and less money. While the displays have gotten higher in resolution, the computers that control them have also gotten considerably faster. This makes a big difference. The speed of personal computers and their displays have consistently been in a neck and neck horserace of increased resolution and the power to make pixels move on it. This is a particularly critical edge when it comes to new media software. As more and more complex calculations are required to fill the display, processors need to be powerful enough to keep animation and interaction fast enough that the viewer doesn't see slow,

plodding media, but has the illusion through persistence of vision[5] that objects appear to move gracefully.

Early tools

In order to work with any computer and create graphics, animation, or interaction you need either a software authoring tool, or you must write your own program in a computer language. In the early days with personal computers these tools were written by small mom and pop industries in their kitchens or spare bedrooms, packaged in plastic bags with a photocopied manual and a single 5-½ inch floppy disk (or sometimes a tape cassette). Usually they were hung from hooks on pegboards at equally small computer shops, managed by computer hobbyists who had little interest in helping novice users. The price of these programs tended to reflect the value they had to the user, often in the $10 to $20 range.

With my 1980 Apple][Plus I had experiences with four graphics and animation oriented authoring tools: Todd Rudgren's (the Rock and Roll musician) graphics tablet software, an animation programming language called (sic) GARPH, a set of odd little graphics utilities from a company called Beagle Brothers, and a very interesting and comparatively expensive program ($128) called Total Graphics System from Accent Software. These pretty much represented the state of the art in graphics software on the Apple][Plus. The only other way to create graphics was to program them in Apple's Apple BASIC language. In fact, this could be looked upon as the watershed moment dividing programmers from computer graphics tool users. Until that time, if you wanted to create a non-text based image on a computer monitor, you had to program it yourself.

There really wasn't any demand for new media authoring software in the early 1980's, which was more likely to be called interactive graphics software. Creating animated on-screen graphics that ran on a personal computer required low level computing techniques that commercially available authoring tools simply weren't capable of at the time. In order to get the speed required, authoring was usually done with very customized, proprietary software with exceptionally unfriendly user interfaces – if there were any interfaces at all.

I was personally involved with two companies between 1984 and 1990 who were, for a time, the only developers of computer-based interactive multimedia in the United States. Accent Software, who had developed the TGS for the Apple computer mentioned above, built its authoring tool with a computer language known as Assembly Language. Unlike other higher-level languages, like BASIC, C, and JAVA, which address an intervening interpretive layer between the software and the microprocessor, Assembly Language directly addresses the microprocessor in terms of 1's and 0's. Since it's difficult for human beings to remember binary numbers, Assembly Language uses mnemonic commands to replace the ones and zeros, so where a binary piece of code might look like this:

```
10110000 01100001
01001111
01001101
```

The equivalent in Assembly Language would look like this:

```
mov  al, 0x61
clra
psha
```

Programmers tend to gasp when you tell them this method was used. But the lack of an intervening interpretive layer between the program and the microprocessor made it possible to do interactive animation in real time. The downside was that every time you wanted to have any program work on a different microprocessor – not just a different computer or operating system – like moving from an 8086 to an 8088 microprocessor, you had to rewrite *all* your code. This factor directly prevented Accent Software from being as profitable as it could have been. It also laid it open to competition as computers got faster and the competitive differential of their proprietary software became less of an advantage.

After six years developing interactive animations for Apple][‘s and IBM PC's I left Accent to go to one of its competitors, SoftAd. SoftAd also had proprietary methods of developing interactive multimedia (the more common name for interactive animation at that time), however where Accent had an Assembly Language authoring tool – a interface for us to draw, animate,

and rig-up interactivity with – SoftAd had a pool of software engineers that took storyboards from the content designers and implemented it in compiled languages like C. In C, you can develop pieces of reusable code, sometimes called libraries, which accumulate as the intellectual assets of a company. SoftAd was the first company to create an asset base for this purpose and could leverage each new project with the assets from previous projects. Pixar and other three-d animation studios use the same principle in creating their tools internally for their movies, requiring competitors to develop them at their own expense.

Authoring Tools Come of Age

Everything changed when a small software tool company, MacroMind, received venture capital funding and moved from Chicago to San Francisco in 1989, founding an area south of market that became known as Multimedia Gulch. Crazy man and force-of-nature Marc Canter had founded them[6] in 1984 when they brought out an animation-authoring tool called VideoWorks (which became Director in later releases) to coincide with the release of the new Apple Macintosh. The Macintosh's graphics orientation made it a perfect platform for such a tool, along with the propensity for Mac users to be from the creative community rather than the business community. When MacroMind came to San Francisco the computing world once again (Accent Software's early consumer version of TGS being the only viable antecedent) had an authoring tool capable of creating animated interactive productions that could be created without a deep understanding of programming in a software language. In 1991 VideoWorks, renamed as Director, was rebuilt to run on Windows as well as Macintosh, further opening its potential for developing media.

Director was a remarkable creative product. Originally envisioned as a tool to create interactive Rock and Roll videos, it was based on a brilliant metaphor: you have a Stage (the computer display) on which actors (Sprites) perform (animate). The Actors/Sprites are kept in a place called the Cast and are dragged (with the mouse) onto the Stage to perform (animate). Their performance is developed and recorded in the Score, and are controlled by a Script (interactive program). Almost all the actions in creating a Movie were done with gestures of the

mouse. The metaphor was easy to understand and users of the tool could be creating interactive animation with only a few hours of training. Compare this to Accent Software's proprietary tool that had dozens of non-mnemonic keyboard commands and took a minimum of six months to gain any proficiency with.

Some other authoring tools provided similar functionality to Director's. Authorware, designed for the educational market, used a drag, drop and link interface to create interactive media. The company that made Authorware merged with MacroMedia in 1991 to create Macromedia. While Authorware was similar to Director, and since the newly unified company saw the wisdom of letting each product dominate its own marketplace, they didn't directly compete with each other.

HyperCard from Apple was an easy to use early implementation of hypertext with additional capabilities in graphics and sound. It only worked on Apple products, which severely limited its commercial viability as a professional development tool. Apple has abandoned it and it no longer runs in their new operating system, OS X. When it was released in 1987 it was a primary influence on MacroMind's (soon to be Macromedia) move from the non-interactive VideoWorks to the interactive Director. HyperCard contained an internal scripting language to control user interactivity called HyperTalk. Director used the concept and much of the syntactical structure of HyperTalk in its internal scripting language Lingo.

Director stood at the top of the heap, essentially alone – Authorware's marketshare was too focused in education and HyperCard's was limited to Macintosh – in the marketplace of tools for authoring interactive media for personal computers until a company called mFactory introduced a revolutionary new tool called mTropolis (pronounced "metropolis") in 1996. It was an object oriented authoring tool built on the same software theories that were taking the software development market by storm with C++[7] – soon to be the most popular application development language used for commercial applications. The CD ROM market upon which Macromedia had built itself was starting to collapse, and critics were lauding the advantages of mTropolis' approach to interactive media authoring over Directors at every turn. Macromedia responded by incorporating many of the features in mTropolis and greatly expanding its power and flexibility.

The competition from mTropolis is, in a way, the best thing that ever happened to Director in terms of its advancing as a creative tool.

The Media is the Message

A peculiar and irksome phenomenon has followed the development of personal computer based new media. As delivery media expands in size, media delivery requirements either expand past the capabilities of the delivery media or if the delivery media catches up, another much less capable media supplants it. For example, when you could finally deliver an interactive experience with 600 megabytes of data at speeds fast enough to play back decent quality video on a CD ROM, the internet becomes the method of choice for delivery and can only allow dial up speed access, a hundred times slower than accessing a CD ROM. Now that broadband internet is common enough to depend on, cell phones have become the preferred delivery format, with slow download rates and ridiculously small screens. It's a constant race Sisyphean struggle to create compelling work that can reach a mass audience.

A New Market, An Old Story

By allowing non-programmers access to relatively quick creation of complex interactive multimedia, Director spurred a new industry: non game-oriented CD ROMs. In the early 1990's the CD ROM market was projected to generate revenues in the hundreds of millions of dollars annually. A confluence of four factors created this market development:

1. The move from proprietary authoring tools to readily available off the shelf software
2. The growth of personal computers for home and entertainment use
3. The decreasing cost of reproducing CD ROMs
4. Moore's Law, which has successfully predicted that roughly every 24 months the power of computer microprocessors will double for the same or less cost. Which plays out on a practical level as exponentially more powerful computers for less money over time.

Many CD ROMs were produced between 1991 and 1995 for sale to consumers in the hope of capturing the projected multimillion-dollar market that had been predicted in the early 1990s. However, the reality of what consumers would be willing to pay for a CD ROM of content and what it took to develop one was way out of sync. It took a team of artist, videographers, designers, even programmers (Director still required some programming skills to work at its best) months of effort to release a CD ROM, which might sell a few thousand copies at best. I knew the end of the market was near when I walked into a CompUSA store in late 1994 and saw a CD ROM on the History of Peanut Butter for $49.95. Who would pay $49.95 to find out about peanut butter's history on their computer, when a full color printed book with the same content would likely sell for less than $25. My revelation was proven when only a few months later I walked into the same store and saw a large bin filled with dozens of CD ROM's on sale for $10 and less.

It's the same old story, the hype of potential revenues builds up an industry that in reality cannot sustain the growth or projected numbers that have been forecast. The decline of the CD ROM market was a tiny little model of what would become the Dot Com Bubble.

Dot Coms and Small Footprints

In December of 1996 a little company called Silicon Beach Software sold its vector-based animation tool that could run in the new-fangled web browsers to Macromedia. Vectors are a method of describing graphic objects as mathematical formulas – lines, circles, arcs – which compress to very small file sizes. I saw the tool when it was being considered for purchase and, frankly, didn't think much of it. It lacked the elegance of full color imagery that I had been used to working with and the demo was amateurish at best. But Macromedia bought it anyway and renamed the tool from FutureSplash Animator to Flash, and a new area of web-based animated, interactive, easy to author content was born. Macromedia had attempted to put Director movies on the web, and they succeeded in doing so, but the software end-users needed to download to make it work, Shockwave Player, was megabytes in size compared to Flash Player's few hundred kilobytes. Even today the Flash Player, version 8, has a small footprint of only 720

kilobytes. This factor, and the fact that movies created with Flash made extensive use of vectors rather than bitmaps, paired with a growing marketplace with slow internet connections, lead directly to Flash becoming the widest spread interactive media format on the web, and therefore in the world of personal computers.

Macromedia rode the Dot Com Bubble to great wealth like everyone else in the sector, and saw its profits and share prices plummet along with the rest. Feeling their Director product would never have the impact they wanted in the web world, and after gambling and loosing on new three-d technology built into Director to boost its presence on the web and its sales, Macromedia decimated the engineering team starting in 2001, reducing it from around 60 engineers and testers to around five by 2003. It is a sad loss to the creative world of authoring new media, with business looking only at expected growth of the growth of revenue year over year in order to justify continued investment in a product.

Statelessness of the Art

Despite the serious Dot Com Bomb, the web is obviously here to stay. It *will* permeate the globe. Pessimists might contend that the rift between the undeveloped world and the developed will prevent the spread of the connectivity of the web to the poor and downtrodden of the world, but I'm a realist. Something as compelling as being able to reach anybody anywhere anytime at almost no cost (and the cost keeps going down and down) will spread even if governments and economics try to stop it.

It has been frustrating as a new media artist and communicator that every time a major form of delivery seems to mature, that is, when it arrives at a point where the technology of delivery and development falls away as the unavoidable subject of creative work in the medium, the game suddenly changes to a new immature medium where the breadth of expression is confined again. But you can depend on the cycle to have an eventual upside of maturity in that new medium, and take solace in that.

In the earliest days of cinema, the Lumière Brothers sent teams of cameramen around France to take footage of anything that struck their fancy. In one example, a camera was set on the back

of a train as it ran through the countryside. It thrilled audiences. But the camera head never moved. It never occurred to them that it should. At some point someone got the bright idea that it would be cool to be able to move the camera from left to right while filming, and invented the pan and tilt tripod head. I look at the state of the art of new media tools today as still in the fixed tripod stage. I believe someone will someday figure out that there might be a new way to approach the tools of the trade, and with a simple extension of capabilities – the pan and tilt tripod head of the 21st century – fundamentally change the nature of the media.

---

[1] Cringely, Robert X.. Addison-Wesley, Accidental Empires, Reading, Massachusetts, 1992, p. 64

[2] DOS, an acronym for Disk Operating System, was IBM's original operating system, before Microsoft Windows took over the PC marketplace in the late 1980's.

[3] IBM wanted to own everything, even the name of the lowly picture element. While the rest of the personal computer world called it the "pixel", IBM maintained for several years in all their literature that it was called a "pel".

[4] It's impossible to be more specific, it was a problem that I and many of my colleges had trying to get a tighter number, since we were dependent upon understanding the marketshare, a particular resolution had at a given point in time, but we could only estimate it based on client and customer feedback. Card and computer manufacturers won't make their numbers publicly available at that level of detail.

[5] Persistence of vision is the trick of the mind that allows us to see film, television and computer animation as moving smoothly as if it were made like the real world. In the real world, when an object moves from left to right, it does so without interruption. In film, TV, and computer animation, the picture on the screen is made of discrete sequential images. Our eyes and brain perceive the movement as continuous. When the change in sequential images becomes to slow, a point known as the flicker fusion threshold (about 16 changes per second compared to motion picture film's 24 frames per second), the illusion that an object is moving is lost and the object appears merely as a changing set of static images.

[6] Mr. Canter is larger than life. It was his vision of a tool to create great interactive rock and roll videos and a deep understanding of what makes C programmers tick. He has a blog at http://marc.blogs.it, and there is a good interview with him that reveals his visionary style at http://www.hotwired.com/davenet/95/36/index4a.html.pre-static.INV

[7] Object Oriented Programming (OOP) is to broad a subject to cover in this paper, but an excellent place to begin is http://en.wikipedia.org/wiki/Object_oriented

Works Cited

Corporation, Quantel. Quantel PaintBox.
http://www.quantel.com/site/en.nsf/HTML/paintbox_home?OpenDocument
Access Date 18 November 2005

Em, David. David Em Bio. http://www.davidem.com/em_bio/em_bio.htm
Access Date 18 November 2005

Evangelista, Bill. No Flash In The Pan. San Francisco Chronical, Sunday, May 5, 2002.
http://www.sfgate.com/cgi-bin/article.cgi?file=/chronicle/archive/2002/05/05/BU175576.DTL&type=business
Access Date: 19 November 2005

Hunter, Glenn. mTropolis, the Greatest CBD Tool that Never Was.
http://www.cbd-hq.com/articles/2000/000501gh_mtropolis.asp
Access Date: 19 November 2005

Oldcomputers.net. The Apple III. http://oldcomputers.net/appleiii.html
Access Date: 18 November 2005

Reimer, Jeremy. Personal Computer Market Share: 1975-2004.
http://www.pegasus3d.com/total_share.html
Access Date: 18 November 2005

Unknown. Beagle Brothers Online Museum: Products.
http://www.panic.com/~stevenf/beagle/products.html
Access Date: 18 November 2005

Unknown. The Unofficial History of Director.
http://www.lingoworkshop.com/articles/history.php
Access Date: 19 November 2005

Weyhrich, Steven. Apple II History. 2005.
http://www.apple2history.org/history/ah06.html
Access Date: 18 November 2005

Wikipedia.org. Moore's Law.
http://en.wikipedia.org/wiki/Moore%27s_Law
Access Date: 19 November 2005

Wikipedia.org. Persistence of Vision
http://en.wikipedia.org/wiki/Persistence_of_vision
Access Date: 18 November 2005

