MYSTIFICATION AND TRANSLATION:

SOFTWARE, ART AND AUDIENCE

Narayan O'Hanlon

Tutor: Chris Fry

2MED614

*

D I S S E R T A T I O N

Course: BA Contemporary Media Practice
Word Count: 9920

Acknowledgements

This document is the outcome of many weeks of reading, writing, waiting and thinking. It is important for me to thank a number of people for their assistance, support, and inspiration over the time I have been researching and writing my dissertation.

Firstly Chris Fry, my supervisor, who has so often guided numerous meandering and tangential thoughts to give me the direction and focus to formulate my argument. I must also acknowledge the support of the teaching staff of CMP at the University of Westminster, who provide their students with a huge wealth of knowledge and experience which myself and the other students are thankful for. My peers and fellow students on the CMP course have given me lots of useful (and not-so-useful) comments, criticism, and entertainment too.

I must show my thanks and respect for the thousands of people and organisations fuelling my interest in this topic: A whole host of artists, institutions, festivals and websites. From them I can always expect innovation and curiosities.

Finally, I say thank you to my family, and Emily, for their help with proofreading, their patience and encouragement.

Abstract

This thesis examines software art from two perspectives: what kind of aesthetic experiences it can offer audiences, and a sociological perspective concerning the relationship between artist and audience. It starts with a contextual study, making sense of the concepts behind software art, finding out why it is, and exploring the discussions about what it is, concluding that vast structures of social science have been assimilated by a software model. This is followed by a discussion of what the aesthetic experience of software art consists of, considering if it can offer new experiences, including some textual analysis. The observations around user experience are framed with analytical, pragmatic and relational aesthetics, from which it is concluded that new aesthetic experiences are achievable, if only as a result of technological and social developments, which software art is a part of. I discover many aspects of the experience of software art lead us to a separation of artist and audience, and suggest ways to reconcile the two.

Introduction

This thesis seeks to examine software art and gain an understanding of the qualities of the aesthetic experiences it can offer audiences, and an insight into the sociological aspects of software art, in particular the relationship between artist and audience. By no means is this thesis illustrative of all prior and ongoing discourse on the topic, my argument is composed of analysis and reflection on other writers' work with my own observations.

Software art, though it has existed in various phases for decades, is still a growing practice. Technologies with new capabilities are in constant production. New hardware, software and user interfaces yield artistic opportunities for innovation. The state of art, particularly software art, is changing and will continue to: This is why it is important to study the artist, the art itself, the audience, and how they pertain to each other. My personal interest in software art stems from having had very poor secondary school education in ICT. I had learned only how to be a consumer of software, which left me somewhat lacking in understanding: Knowing what to do with a computer, but not knowing what the computer itself does. From my own practice in video installation, I realised I had very limited control over how to trigger and manage my own work, unless I could learn how to write my own software. Through open-source hardware and an IDE I began to prototype and program simple projects of my own, leading to my interest in the theoretical concerns discussed in this thesis. Initially, my subject area of interest was too large–the definition of art and artists, from a Poststructural and Postmodern perspective. A lot can be said and proposed for this topic, but after initial research I found myself drawn towards the sociological aspect: The role of

the artist and audience, but also the importance of the artwork itself and whether software art can offer a new experience or only relay those of other media.

The first chapter focuses on explaining the lead-in to the situation in which I write my thesis. I establish the historical and political narratives that inform the basis of my writing, and also discuss software art in the incarnation we see it in today. I start by investigating computational development and how this field has grown to have a bearing on the fundamental operations of society. I look at the role of the artist and the intersection of art and computational development. Rather than outlining every single historical event, I focus on the evolution of the concepts surrounding software art and also explore a variety of interpretations and implementations of it.

In the second chapter I ask what kind of aesthetic experiences we can get from software and New Media Art and whether this experience is something new. To frame my analyses I consider analytical, pragmatic and more recent ideas of relational aesthetics. Starting with a reflection on the ideas of Dewey, I discuss what might be the current aesthetic experience of software art in regard to its exhibition, distribution, material quality, impact, and aura. I give particular attention to interactive art and non-art software, emphasising the user/players experience with reference to ongoing discussions about enchantment in human-computer interaction.

The third chapter takes a sociological look at contemporary art practice, and addresses some of the difficulties facing contemporary artists today. I elaborate on a range of factors and situations unearthed in the second chapter which illustrate the re-emergence of an esoteric artist, and suggest ways of combatting this. Using Bordieu's ideas of class as a backdrop, and elaborating with the ideas of modern art disparager Spalding, I depict a divide between artist/programmer and

audience. The argument continues by showing how this divide is perpetuated by a number of wide-reaching factors on the sides of both high and mass culture, and the space in between. The overall argument of my thesis will be developed by analysing and considering texts and cultural artefacts, to form conclusions about the current state of art and providing a basis of understanding with which we can think about its future.

Chapter 1:

The emergence of the concepts underlying software art.

To look at software in an art context we must understand there exist two contrapuntal strands that are the foundations of my discussion. My thesis is based on the point where art practice and computational development converge, so I will first highlight some of the relevant moments in these two narratives. The seeds of a 'digital way of thinking' were sown when logic was proven to be not just a philosophical discourse but a mathematical one. Mathematician George Boole's system of binary logic converted the notion of logic into algebra, a system that when applied to relay circuits went on to be the foundation of modern computers; his ideas are fundamentally intertwined with the digital and the way computers operate today. This digital way of thinking went through its phase of development in mechanics with creations like Jacquard's Loom streamlining production methods and taking the reins where human hands were more limited. The scope of the early machine computers reached far into the 20th century, with the punched card method of programming the Loom still being a widespread method used for programming and data storage in digital computers as late as the 1970s. These early developments fundamentally contributed to capitalism and the division of labour too, mechanising and ultimately digitising workers and the systems therein: Converting people and commodities into interchangeable and quantifiable pieces. Computational development facilitated a wider digitisation of society, with overarching periods of modernism and futurism envisioning and celebrating the city as a self-propelling machine, through to the Second World War. That, the Cold

War and the following years saw a great inciting point for technological development, as the ripples of the funding for weapons and communications systems would carry through the following decades. This was not without criticism: Nicolas Bourriaud (2002) explains a battle between two schools of thought, "a modest, rationalist conception [...] and a philosophy of spontaneity and liberation through the irrational, both of which were opposed to authoritarian and utilitarian forces" (p. 12). Whether one sides with the Rationalists or the Surrealists he explains that rather than achieving the outcome of emancipation,

> *the advances of technologies and 'Reason' made it that much easier to exploit the South of planet earth, blindly replace human labour by machines, and set up more and more sophisticated subjugation techniques, all through a general rationalisation of the production process (ibid).*

Theodor Adorno also cautioned against the rapid advancement of technology at the expense of liberty. He questioned the rationality of reason in a society where

> *the progress of modern science and medicine and industry promise to liberate people from ignorance, disease and brutal, mind-numbing work, yet help to create a world where people willingly swallow fascist ideology, knowingly practice genocide and energetically develop lethal weapons of mass destruction. (Zuidervaart, 2003)*

He accused science of leading people towards a totalitarian society, as its research and funding is tied up in capitalism and producing 'more' and 'better' ways to ultimately pursue profit. Money itself has over the years seen its materiality be fully digitised. At first there was the abstract concept that any two disparate commodities can be compared by their 'monetary value' for which they can be exchanged. This translated into money itself being commodified, eventually leading to it being removed from the materials 'of value' it originally represented: Money had become 'semiotised'. Finally the physical manifestation of money is fully removed by e-banking and credit cards. Each step benefits the whole capitalist system as it allows greater movement of value through credit, with stock

markets trading using 'futures'. "We have entered into a terrain where software development and [...] 'software culture' have become significant aspects of our contemporary social and cultural life" (Broeckmann, 2006). We can see that the notion of the digital isn't just something to be found in computing, but has become synonymous with so many of the artefacts of late 20th century culture.

The role, responsibilities and expectations of artists have experienced major rites of passage over the last century, too. The Renaissance had set the standard of highly skilled, enigmatic artist-creators, whose esoteric techniques made celebrities out of them. The Baroque, Rococo, and Romantic periods continued to establish these ideas through to the mid-1800s, when the products of the Industrial Revolution pushed towards a critical stance on the Enlightenment and existing grand narratives, eventually spawning many movements of non-representational art. One of the most-cited moments in this sudden change of pace in the art world is Duchamp's 'readymades'. His placement of an existing 'non-art' object into the established gallery space showed that an artist could take up new, political roles through their practice. It asked the now platitudinous (but fundamental) question 'what is art?', this question has been renewed time and again through postmodern art practice in the late 20th century where artists have attempted to shock or create extremities of what could have been considered art. We must discuss this fundamental question again in the frame of software art, as it is as relevant as ever now. It became possible to establish software art in its own terms as it became a well-trodden path. "Software art experienced a brief period of intense attention between 2000 and 2005 [...] software-based artworks now enter into contemporary media and art shows with greater ease than they did ten years ago" (Broeckmann, 2006). Today, festivals like transmediale and Ars Electronica are dedicated to the sharing and understanding of Software as an artistic medium.

The former ran four yearly competitions, with the first in 2001, to explore the "cultural understanding of software" (ibid). After a long history of the underlying concepts of software art being explored, we have only relatively recently been able to reflect on what we consider today to be part of the 'genre'.

The notion of software as art can be understood in different ways. On one level, the two concepts are comparable in structure: Art is something ethereal and aesthetic found on a physical art object, and to compare, Software and Hardware share those respective roles. Software itself is in a way, autonomous, it can run by itself on any (compatible) machine. To substantiate this, we can understand that art follows a triadic relation: Once the artist completes a work it only has a potential to be art, a potential which is fulfilled by an audience, otherwise it is only an 'object'. "Art is complete only as it works in the experience of others than the one who created it" (Dewey, 1980, p. 106) This observation works in an obvious way with many kinds of New Media Art too, more often than not requiring interactivity to be realised. The distinction we can make between more traditional art forms and software in general is that its hardware can remain unchanged but take on different software which uses a common language. Jack Burnham describes software as "an attempt to produce aesthetic sensations without the intervening 'object'" (cited in Shanken, 2002, p. 434), and in recent times, having seen technology become in some ways ubiquitous, the statement is as relevant as ever. One fact of the matter though, is that when talking about software art, we will need some kind of hardware on which to execute it, so it seems it cannot produce aesthetic sensations without some kind of coding and decoding go-between, a computer. After all, humans are analogue beings. As previously discussed, computing and the language of the digital are structurally derived from analogue machines, so software, while synonymous with these spheres does have its parallels

in non-digital, 'organic' processes. Sol LeWitt's conceptual instruction-based art and 'wall drawings' were simply ideas and plans to be made reality by groups of other people. The Fluxus Movement's Events "presaged many current digital art concerns" (Gere, 2008, p. 80) and are in some ways parallel to software, in the sense that a series of instructions, like a thread, are to be processed by the hardware of the body and other objects. Work by this group and other artists "reflected the concerns of a world in which information and communications technology [...] were becoming increasingly important" (p. 79), while not necessarily involving computers in their work it usually was necessary to have some kind of feedback loop - "the active participation of a receiver as well as a sender" (p. 89). A contemporary example of these kind of 'Events' could be David Horvitz's 2009 email group. Horvitz sent a set of instructions sent to participants on a mailing list every day, to be performed and shared. Sometimes his instructions were mail-art based, a practice parallel to executing a piece of software: Appropriating an existing distribution structure on which to base his project. Horvitz's projects rarely required the participants to use digital technology, other than to receive the instructions - the participants being physical extensions of the hardware. Fluxus and Horvitz are just two examples of artists using a non-digital approach but whose practice has parallels to software art. Despite Burnham's assertion of there being no necessity for an 'intervening object' in producing aesthetic sensations, it is difficult to suggest that this type of art exists in the ether, notwithstanding its autonomous and ethereal nature. The configuration and presentation of the art's hardware is one method of defining the variations and subcategories of software art. One example of such a category is Internet art: Since the 1990s, there has been a proliferation of digital art, particularly with the advent of Internet art and Telematic art in which there is a practical application of

networks and usually involvement of an online audience, dispersed throughout the world. Another variation is Algorithmic and Generative art. These processes can be used to highlight the autonomous nature of software which could mean non-interactive screen-based Installations. We need to look not only at the medium or the 'hardware' used in a piece of art to define what (sub)category it sits in but also the concepts that tend to be discussed. Currently these are usually inferred by or inherent in the nature of the systems being used, for example the use of public networks, surveillance or communication systems automatically raises topics of ethics, generally. JODI.org's (wwwwww.jodi.org, n.d) 'browser art' approach is a quite postmodern one, using the digital language as their medium and a user's own web browser as a gallery they present a labyrinth of hyperlinks, remixes of existing 'mass' software, ie. video games into a digital glitch-bricolage, snippets of programming language and humorous legal disclaimers. Their approach, through reflexivity, draws our attention to the medium and shatters the illusion of transparency we experience from most user interfaces. The simple fact that their work is using the web as its distribution method hampers the work with a whole host of pre-set conceptual themes. With time these technologies and practices may be commonplace enough to not have to automatically infer themes; we do not see a photograph as having to always deal with the theme of 'light', it is just part of what defines the medium, an artistic tool, not necessarily the intended narrative presented by an artist. We can see that there is a lot of scope conceptually and in terms of hardware configuration for these different kinds of artwork, thus we can draw that 'software art' itself is actually quite a vague term. runme.org, an online repository for software art provides a list of software art categories including 'code poetry', 'browser art' and 'conceptual software: without hardware - formal instruction'. As this database is a culmination of many peoples cumulative

understanding of a fairly new term, it is in a good position to define it. The site's 'about' section provides a summary:

> *Software art is an intersection of two almost non-overlapping realms: software and art. It has a different meaning and aura in each. [...] Software culture lives on the Internet and is often presented through special sites called software repositories. Art is traditionally presented in festivals and exhibitions. Software art on the one hand brings software culture into the art field, but on the other hand it extends art beyond institutions. (runme.org, n.d)*

This careful approach establishes distinct differences between more traditional forms and software. As Edward Shanken (2002) points out in his paper 'Art in the Information Age' presented at SIGGRAPH, "art-historical literature traditionally has drawn rigid categorical distinctions between these practices [of conceptual art and art-and-technology]" (p. 433), but as we see more examples of art joining these once disparate practices, we start to understand the notion of art-and-technology to be a part of "larger social transformations from the machine age of industrial society to the so-called information age of post-industrial society" (ibid). Marshall McLuhan had a suggestion to this effect, that "art was a distant early warning system that can always tell the old culture what is beginning to happen to it" (cited in Gere, 2008, p.16). If this was to be true at the time of writing, with Fluxus et al presaging the concerns of a technology-saturated Western world which we have seen emerge in the time since, then we should be able to trust that with hindsight, the software art of today will have been speculating and anticipating the coming decades too.

Chapter 2 :

To what extent does software art present the opportunity for new aesthetic experiences?

In 'Art as Experience' (1934) John Dewey presents his argument of how we understand an aesthetic experience. As art and aesthetics have changed conceptually and materially in the 75 years since the book's first publication, I will try to update and highlight the relevant ideas. Dewey describes art as being proof of "a realized and therefore realizable union of material and ideal" (p. 27), the harmony of which allows the aesthetic experience. Despite his acknowledgement of 'technological art' (p. 47) he claims that "every art does something with some physical material" (ibid), and while I cannot fault Dewey for not being a soothsayer it is clear in the frame of my argument that art can be conceptual, digital and even completely immaterial. Art was the rope in a tug-of-war between 'material' and the now victorious 'idea'. By this analogy I mean that we now have the option of non-material tools and media with which to articulate our ideas, the ideas being reasons for the existence of artworks in the first place. Art in general is encoded, in that the meaning can be drawn out by analysis and engagement with the art. In particular with software art, this statement has literal and reflexive layers too. "Each medium says something that cannot be uttered as well or as completely in any other tongue" (p. 106) suggests there are multitudes of languages in art, each specific to the medium. The statement could be true of software art, as discussed earlier there are certain established recurring thematic elements that this medium is naturally disposed to. This implies that software art as a relatively new

medium presents an opportunity for well-polished ideas conceptually, but can that mean entirely new aesthetic experiences?

It is necessary to discuss what an aesthetic experience of software art consists of. The context in which the art is viewed changes how an audience feels about the work, so the fact that software art can be (re)produced or run on an individual's personal computer or even a mobile device can make a work of art personal, in a private sense. This leads to the question of how the art operates and occupies the device - it may be that you can be linked to the work by somebody else, and that it works within the confines of the familiar UI like a web browser (jodi.org being an example of this). An audience member can be alone with the art, and perhaps control it in an intuitive way. The way that Internet-based art can be shared like this, for example through platforms like runme.org, gives the possibility for potential new audiences to be involved and engaged with art, no less because it is almost always the case that software art is interactive. Interactivity provides a feedback loop, which in the case of software art is unique, in that it helps an audience to see the art in its various iterations - its meaning can change on the fly. This opens up possibilities in terms of having narrative or time-based concepts, and taking advantage of that aspect of the medium. It gives users a feeling of communication, of life or intelligence, albeit artificial. It is that very mystery, or beauty, that fundamentally legitimises it as an artform, "it saves it from being mechanical. It gives the spontaneity of the unpremeditated" (Dewey, 1980, p.138). This allows for both the artist and the audience to revisit the work, add to it, or to make different choices when interacting with the art, adding milage. It can only loosely compare to revisiting a painting in a gallery, a distinctly different experience. Other types of software art that do not follow this exhibition/ distribution method will conform to a different set of rules. If the work is gallery-

based and involves installation, then a whole number of traditional ideas about the notion of the aura are brought to the table. The aesthetic experience is clearly affected by this environment, but is any new experience offered? While the subject matter may be contemporary, the audience ritual of looking at 'caged' art in a gallery tends to remain. Art in this context may fall somewhere else under the 'New Media Art' category, though undoubtably lots of digital artworks use software to some degree, as a methodology or theme. The fact of the gallery though, can limit the scope of a piece. A recent site-specific public art installation by Mark Dixon, unveiled on a new housing development in Orchard Park, Cambridge made use of the democratising potential of telematic art. A large antenna-like sculpture, adorned with flashing LEDs that were triggered only when mobile phone signals travelled through it. This became something of a phenomenon for the locals who engaged with it, directly or incidentally, on a daily basis. We could say that there is something in this human-computer interaction that enchants the audience who may have little experience of this kind of art. It is partly the remit of public art, but I believe it is within the remit of software art too, due to its pervasive and 'democratic' nature, to engage the audience with open interactions like this.

Nicolas Bourriaud's theory of Relational Aesthetics can be used to describe these types of interaction between art and audience. He believes that there is a new role for art and architecture - "no longer to form imaginary and utopian realities, but to actually be ways of living and models of action within the existing real" (2002, p. 13). This offers art the opportunity to be more substance than style, or to have art where beauty and meaning are brought by the audience from the outside. With this approach to the 'sharing' of an aesthetic experience, we see a

break from the orthodox. Jack Burnham illustrates this–here in regard to Conceptualism, though the same could be said for Relational or software art:

> *Through the history of art there have been certain tacit relationships between dealer, audience, collector and artist, establishing degrees of control over the production and dissemination of a work of art. These however break down with Conceptualism, and part of an art- work becomes the assignment of new control variables over the life duration of a piece. (1999, p. 217)*

Installation artist Rirkrit Tiravanija describes people as having utmost importance to his art - "without people, it's not art–it's something else–stuff in a room" (Bishop, 2004, p. 61). Bourriaud believes that we can experience Relational Aesthetics simply by taking part in artworks that offer this kind of interactivity. A critic of Bourriaud, Clare Bishop argues that it is not simply about taking part, and that we must be accounting for what kind of quality of audience relations the experience gives. Relational Aesthetics is a concept very much present in most software art, particularly telematic and internet art that use live audiences, as this approach falls within "the realm of human interaction and social context" (Bourriaud, 2002, p. 14). With the ideas of Bourriaud and Bishop, there is a necessity to the physical reality of the art, the human-to-human interaction is important, which is somewhat of an oversight of software art, which holds the computer as instrumental in the interaction. Nevertheless, relational art as a concept is very relevant to my discussion. Bourriaud claims that relational art, presented in a gallery or exhibition space "tightens the space of relations, unlike TV and literature which refer each individual person to his or her space of private consumption" (p. 16). Interactive art exhibited online can also have this effect while allowing individuals their privacy. This method of exhibition could indicate some scope for new aesthetic experiences, and actually highlight Bishop's point when she says that "relational art sets up situations in which viewers are not just addressed as a collective, social entity, but are actually given the wherewithal to

create a community" (p. 54). It is obvious that the network of the World Wide Web has huge scope for this kind of action and there is much evidence for online communities, therefore Bourriaud's insistence on in-person relationships seems very limiting to what can be deemed relational art. There is truth in Bishop's claims that "contemporary art solicits the viewer's literal interaction in ever more elaborate ways" (Bishop, 2004, p. 77), and is about "fulfilling the artist's interactive requirements" (ibid), but in the context of non-digital art, she presents a criticism of this. A more positive outlook on these attempts by artists to engage an audience with literal interaction can be found in the realm of software art, which by its nature is designed with literal interactivity, between human and computer, in mind.

Another angle on aesthetic experience is the idea of *enchantment* in human-computer interaction. Enchantment, an idea studied by John McCarthy, can be described as a feeling which provokes "affective attachments to particular interactive systems" (McCarthy, 2005, p. 370). Interactive systems which I have experienced enchantment with are some recent 'sandbox' games. With this type of game there is some responsibility for the player to create their own situations and narratives. One such game is 5th Cell's platform game Scribblenauts. By typing in nouns, the player can create objects and characters which interact with each other in different ways, and can be used to solve challenges. For example spawning an angry god and an atheist will cause them to fight, or putting bread in a toaster creates toast. The scale of the library of nouns and adjectives is massive, so players tend to simply test its limits. The game follows the rules set out by McCarthy when he writes "an object or interactive system that is likely to evoke enchantment should offer the potential for the unexpected, give the chance of new discoveries, and provide a range of possibilities" (ibid). Obviously the possibilities given by

these kind of experiences are finite, limited to a dictionary, but players can use the tools they are given by the developer to create or generate something which pushes the boundaries outlined by the developer, fuelling their curiosity. Garry's Mod, a modification of Valve's Half Life 2 gives players the opportunity to spawn and arrange objects and characters found in the main version of the game. By providing this platform, the developer unlocked the ability for players to make their own Machinima films, prototype mechanical creations, or create new challenges for themselves. This kind of software goes some way to removing concerns of an audience "fulfilling the artist's interactive requirements" (Bishop, 2004, p. 77) because the sandbox elements give the game a certain depth, rather than a cause-and-effect interaction. One can also get a sense of enchantment from the hardware itself. Apple's iPad can run a vast and complex game like Scribblenauts, and just as McCarthy said about the G4 Powerbook, the iPad itself "challenge[s] assumptions about what computing is and can be and particularly about relationships between aesthetics and function [...] evoke[s] space travel, weightlessness, artistic creativity, and even television or cinema screen culture" (2005, p. 370). Though McCarthy uses commercial examples of technology providing enchantment, the explanations McCarthy gives us are less specific: "Enchantment [is] an experience of being caught up and carried away [...] perception and attention are heightened [...] it awakens us to wonder and to the wonder of life, it is enlivening" (ibid). It is clear this can carry over to digital artworks and immersive installations too.

Recently McCarthy's idea of enchantment has been strongly evident in the users of personal computers and the operating systems/software therein. Icons, windows, touch screens and gestures being part of a user interface are all attempts to give transparency to the front-end of software. To compare this to what we suppose to be software art we can see uses of analytical and pragmatic aesthetics on

a number of levels. Firstly there can be inherent 'classical' beauty in a UI in how closely an interactive experience with it resembles a real-life interaction with an object (its transparency), evidenced by the demand for graphical interfaces in computers, touch screen devices, and the enchantment, or technological reification that happens in the mind of the user. "The combination of emotional attachment together with a sense of something not yet understood leaves us feeling disrupted but also alive, attentive, and curious" (ibid). This enchantment that McCarthy writes about relates strongly to analytical aesthetics, as the aesthetic beauty itself is assumed to lie somewhere in the coding, or perhaps in the graphical elements of the interface itself. In general there is some naivety in analytical aesthetics that a pragmatic approach tries to deconstruct. To take a post-structuralist standpoint (eg. Dewey), beauty itself is a cultural construction, and the aesthetics of software do not necessarily originate in the way they are coded, but rather they exist between the object and the reader, in the act of reading: It would be rather difficult to apply analytical aesthetics to software, the suggestion being that bare code contains all the aesthetic properties of the work. With pragmatic aesthetics in mind we might be able to make such claims: Geoff Cox, Alex McLean and Adrian Ward do just this in their essay 'Coding Praxis'. They argue that "code may have aesthetic value in both its written form and in its execution" (2004, p. 161), drawing comparisons of written code and executable software to written and spoken poetry. They use the example of 'Feedback', a piece self-modifying code written by McLean which, as it runs, keeps changing its own source code. The work is generative and self-propelled, it gives its own performance–code that *is* art. On the other hand we have code that speaks in unison with its own execution, Scott Snibbe's 'Tripolar' is a java applet which draws lines directed by Newton's laws of motion. A line commented out in the source code reads "// The source code

demonstrates the "meta-chaos" of the program itself" (CODeDOC, n.d), the programmer himself acknowledging a harmony between the 'written' and 'spoken' versions of his work. Cox et al align with Walter Benjamin in admitting that "there is some danger of the aestheticisation of code at the expense of other factors" (2004, p. 167), referring to Walter Benjamin's concerns over "the introduction of aesthetics into political life" (1935, p. 14), but rather than leading to war as Benjamin claims, they may simply be suggesting that there are certain practicalities that need to be considered when creating a functioning piece of software.

In aesthetic discourse, Benjamin's writing around the aura of an artwork 'The Work of Art in the Age of Mechanical Reproduction' (1935) is much cited. In the search for new aesthetic experiences we must revise this idea, as nearly 80 years of art practice which discusses and challenges his ideas have passed. The basic premise is the undermining of the aura of an artwork in order to counter existing aesthetic values: "The technique of reproduction detaches the reproduced object from the domain of tradition" (1935, p. 3), and this process still exists today with new media. What has changed is that 'existing aesthetic values' are difficult to pinpoint because of the paradox of an aesthetic zeitgeist when our collective consciousness as a society has become so fragmented, rather a bricolage of our individual subjective thought. Software art can undermine the aura in a way, but we must establish first where the aura lies, or how it is experienced. In the past the aura of an artwork has stemmed from its authenticity, its lack of reproducibility - this kind of "bogus religiosity" (Berger, 1972, p. 23), at once, *is* and *isn't* applicable to software art. By design, software is reproducible; a line of code remains unchanged when running on any system, and its distribution is potentially dispersed. Benjamin's observation that the camera makes reproduction of artworks

possible – with the caveat of the loss of authenticity, is acknowledged and updated 40 years later by Berger (1972) in 'Ways of Seeing'. He concludes that the 'religiosity' we find in supposedly authentic artworks is bogus because "If the image is no longer unique and exclusive, the art object, the thing, must be made mysteriously so" (p. 23), and that this constructed idea of authourship and authenticity produced by a gallery setting is "the substitute for what paintings lost when the camera made them reproducible" (ibid). Gallery-based New Media artworks which use software as an element can still evoke this traditional aura described by Berger for two reasons: Firstly the situation of the artworks will command it, as galleries do. Secondly the interactive element, the actual aesthetic experience of New Media art in galleries or expanded work is difficult to reproduce with photography or video. One proponent of aura in New Media Art is Anne Lawrence. In her essay 'The Spatial 'Aura' of Mariko Mori's Pure Land', she makes a case for the aura in new media installations. She argues that the spatial concerns of a digitally created installation can provide an aura that "is completely up to the artist to create and maintain. [...] Once the entirety of the installation is replicated, time and space will no longer be unique" (Lawrence, 2000). In the enthusiastic way Lawrence describes Mori's installation, McCarthy's explanation of enchantment springs to mind: "caught up, carried away [...] perception and attention are hightened" (McCarthy, 2005, p. 370). I would hasten to suggest that enchantment and aura are the same thing, but HCI enchantment as a concept seems to be closely related to the notion of an aura, within the realm of analytical aesthetics. To refute Lawrence's claims that a Benjaminian aura still exists 'spatially', the three dimensional spatial concerns which are so key to her argument are far removed from the spatial concerns used in Benjamin's time, yet Lawrence claims that it is the *same* aura that she is discussing. They are not talking about the

same spatial qualities. To agree with Lawrence is to void the possibility for a *new* kind of aesthetic experience of the 'aura' for New Media Art. If we focus on non-spatial software art online, replayable and perfectly duplicatable, we are forced to overlook even this argument and must move nearly another 40 years forward from Berger. Seeking to use an analogy parallel to Berger's, it could be said that there is no 'religiosity' in internet art, because of the removal of the religious institution, the gallery. Instead what substitutes the religiosity is the hyper-mystification of the artwork. Neither Lawrence's or Benjamin's definitions fit this mould, perhaps for this type of artwork we can declare the aura well and truly 'withered', but this is not to say users can't have engaging interactive experiences –literal and mental, with them.

Quayola's video installation series 'Strata' is an analysis and example of the idea that aesthetics exist in a pragmatic and subjective way, deeper than just the surface of an artwork. Using images of Baroque and Renaissance paintings as his starting point, he created videos in which unnatural polygons start to form on the surfaces of the paintings, picking out subtle geometrical patterns. The shapes move and dance with the rhythm of dark electronic drones. By augmenting the 'classical' beauty of the paintings he manages to create an alternative aesthetic experience. The piece unearthed new ways of seeing the old paintings by using generative software. When looking at the painting, an audience can observe the brushstrokes, colours, shapes, and from this imagine the process of creating the artwork. This is not yet the case for software art. For most, the aesthetic experience of software art has a removed appreciation, no ability to relate or understand the creation of it, no less because the 'brushstrokes', are in some ways hidden. Given paper and a pencil, everyone can make marks that shows or represents something, it is just a question

of skill. Given programming tools, proportionately very few people will be able to make anything coherent.

Enchantment in human-computer interaction and the lack of knowledge about the workings of software both serve the function of mystifying art and its processes, as McCarthy describes an "antagonistic relationship between knowledge and enchantment" (McCarthy, 2005, p. 371)–if you understand the magic trick, it doesn't astound you. David Gauntlett (2011) believes that "through creative activity, where making is connecting, we can increase our pleasure in everyday life" (p. 8). The assertion one could make that 'Art exists to enrich peoples lives' can be generally agreed on, and in addition, life-enriching experiences can be results of the enchantment that may come from the transparency of an interface. Unfortunately the whole notion of a 'transparent' interface is an illusion (an idea which shall be elaborated on in the next chapter), and aids in the alienation of art: it "does not follow at all the road of demystification and unmasking, but promotes a hyper-mystification" (Perniola, 2004, p. 49), one inevitability of this mystification of the art, and its production, is the re-emergence of the esoteric artist.

Chapter 3 :

Changing attitudes: Audience and artist.

The dispute between proponents of high art and those of mass culture is a longstanding one, and for Bordieu, an issue of class: "art and cultural consumption are predisposed, consciously and deliberately or not, to fulfil a social function of legitimating social differences" (1984, p. 7). It is important to acknowledge these debates when looking at software art and relational aesthetics, as these power structures exist on various levels. The vast majority of the population have no expert experience or understanding in the fields of software art criticism or practice. The primary way people experience software is through apparent 'non-art' programs: desktop and mobile operating systems, games, commodities etc. In 'The Eclipse of Art', Julian Spalding comments on the difficulties the public have with so-called 'Modern Art', primarily that it has become increasingly inaccessible and "could be heading into a cul-de-sac" (2003, p. 9). He agrees that "the art of our age [exists] in the dark, the self-indulgent plaything of a few" (ibid), echoing the concerns of Perniola. The art world has internalised its discourse so that firstly "a rift has opened between the art being promoted in contemporary galleries and the art people like to hang on their walls at home" (p. 11), the fact of which is off-putting and alienating to both parties, and secondly he comments on New Media art "which people used to the traditional languages of art [...] find difficult to comprehend" (ibid), which serves to divide these communities even further. In his essay 'That Withered Paradigm' Peter Walsh highlights the opinion that an 'Expert Paradigm', which he describes as "information hegemony" (2004, p. 1) is under threat. It is apparently challenged by the freedom to communicate information in

new ways, namely the Internet–just as Martin Luther's use of printing on a large scale had challenged the knowledge hegemony of the Church, thus kickstarting the Protestant Reformation (p. 2). A recent example of technology significantly challenging these hegemonies can be in 2011 the use of social networking site twitter to break 'super-injunctions' that prevented established news outlets from reporting on certain stories. Because of these kinds of events, Walsh comments that the paradigm of the expert has 'withered'. This is a significant argument, as can be used to dispute my assertion that we are seeing a re-emergence of the esoteric artist. The historical examples used by Walsh show that technologies that challenge knowledge hegemonies, do not destroy these hegemonies but rather force the institutions or individuals to re-evaluate their path. Walsh concludes: "It is unlikely that the Web will destroy the Expert Paradigm. It will, however, continue to alter existing paradigms, to push them in new directions and into new forms" (2004, p. 4). I agree that the Expert Paradigm exists and will continue to, but to address the knowledge hegemony being dealt with by this thesis– that of software art production, we can see that the internet has already equipped software artists with more tools to grow their hegemony, and it seems the only way to break free of this is somewhat unrealistic: To saturate the world with the knowledge another way, namely teaching software programming to all, through schools. Spalding laments the fact that artists "are born and unmade. [...] As one goes up the age range, art disappears from the education system" (2003, p. 51). Art classes have taught the classic media for decades, to update this would enrich the individuals, but also society and the economy in the long term. In early 2012, the Department for Education in the UK was "looking to industry, organisations and learned societies to help build a replacement curriculum in computer science" (Livingstone, 2012) for Secondary school children. Janet Murray (journalist) claims the inciting point

for this was when Google's Chief Executive Eric Schmidt "publicly attacked the UK for failing to capitalise on its record of innovation in science and engineering, saying the country that invented the computer was throwing away its 'great computer heritage'" (Murray, 2012). Many concerns about the knowledge hegemony in computing will perhaps be addressed by this, at least in the UK in the coming years. The importance of teaching in this area is not to simply tutor students in programming, but to perhaps help the young understand and engage with the 'digital world' in which they are growing up in: "Computer science is not just about programming, it's about computational thinking, problem solving, analytics, physics and creating code. Building digital content and intellectual property. Building value in the digital economy" (Livingstone, 2012).

To return to Bordieu's sociological approach, the social difference maintained by art and cultural consumption may be challenged, in regard to software as art, as the knowledge and practices become widespread. As already discussed, the World Wide Web is one potential platform (or rather, plays host to many platforms) on which to spread information and break down knowledge hegemonies. Having knowledge of interactive systems also allows us to access the secrets of the 'Magic trick' of enchantment. When contemplating information hegemonies, magic is a pertinent analogy for enchantment– it is implicit that Magicians do not give their secrets away. If the knowledge of programming and software development becomes widespread, then despite living in a world of ubiquitous computing where technology has assimilated social processes, as humans we should be aware of these surroundings and their workings–we should be able to humanise ourselves with our understanding of machines. There is some irony in this suggestion insomuch that the more we become aware and knowledgable of computers, the more we are able to escape being 'sucked into the

machine'. Current online networks are doing quite the opposite: They allow the subjective to become fused with the digital, giving users an experience parallel to reality. On any consumer computer it is easy to buy something, communicate with words and pictures, and experience art at the same time, and to go between these experiences fluidly, at will. Virtual reality is an experience offered to these consumers on a daily basis, no less with online 'worlds' like the much discussed 'Second Life' (PlayStation Home, There, Onverse and many more all offer the same basic experience). They attempt to parallel real-world structures like economy and culture, and yet cross over with real economics and cultural concerns. Inevitably, they also raise questions of authourship – who is responsible for the creative output of a piece of software? SL plays host to a range of 'galleries', as well as providing 3D modelling tools for players to create sculptures, landscapes, clothes. Though a participant would not be right to claim authourship over the software, can a developer claim authourship over a user's actions, which are a technical eventuality of the developers work? Businessweek ran a cover story in 2006 about a Second Life character worth over $250,000 (later going on to over $1M), and a small online empire "so strong that it now has to import skill and services from the real-world economy" (Businessweek, 2006), meaning the hiring of a team of people to manage assets in the virtual world, using their own avatars. This poses the question: Can we accept the virtual space that software creates, as being as good as real? One angle is that we *want* these worlds to be real–to accept the virtual as real is to give in to a psychological projection. In 1988, John Walker advocated a transparent and utopian vision of human computer interaction engulfing the user, claiming we need "to move beyond the current generation of graphics screen and mouse, to transport the user through the screen into the computer" (Walker, 1988), demanding the embrace of ubiquitous computing, reiterating my earlier

assertion of being 'sucked into the machine'. Using the analogy presented by a computer's operating system–a 'window', one could suggest that screens we use today are windows, letting us look *through* into an illusion of three-dimensional space. Jay Bolter and Diane Gromala comment on this premise of "looking at and looking through at the same time" (2003, p. 34) in their book 'Windows and Mirrors'. They present a criticism of the transparency that user interfaces, games, and software art alike all tend to conform to, claiming transparency is a fallacy, or illusion. They use the example of Daniel Rozin's interactive installation 'Wooden Mirror' to show that it is possible to create digital art that is transparent *and* reflective, allowing a participant to see 'through' a surface but also reflecting them and drawing attention to the medium. The aforementioned online 'virtual reality' worlds are comparable to techniques of perspective painting: "Paintings, like digital applications, offer an experience, and perspective painting offered the same experience as the one now promised by virtual reality–the experience of "being there"" (p. 36). In this regard, photography usurped painting, which to survive aligned itself toward non-representational methodologies, often denying the viewer the access to virtual depth and instead highlighting the qualities of the medium itself. Bolter and Gromala present designers with a choice–once the 'history of transparency' is understood, it is down to individual creators to engineer their creations as windows or mirrors. "Painters do not have to aim for transparency and neither do digital designers" (p. 35). It is true though, that the majority of people, the 'markets', want transparency in technology and artificial life/reality, because it is easy. The risk this path runs is one of the division of knowledge, and perpetuates an oligarchy where there are few who have a wealth of complex knowledge and use it to pacify the many, mindfully or inadvertently. Is it possible to reconcile the authour and audience in this situation? One approach we

could use is that of David Gauntlett's, to encourage creativity for all. On one hand, the stimulus may be a case of politicising or educating a public that "have become comfortable with the undemanding role that contemporary culture expects us to enjoy" (2011, p. 7), and on the other hand make sure that there are tools available for creativity that are "as open and inviting of creativity as possible" (p. 6), but not limiting: "Tools which only offer a predetermined set of opportunities [...] deny creativity and impose the fixed meanings of others" (ibid), therefore sustaining the knowledge hegemony. Another method we can use to reconcile the parties is laid out by Spalding, and requires action from the audience rather than the artist:

> *Increasing your experience of art is the best protection against being deceived by appearances. This is [...] learning how to respond to each more fully and deeply. Looking at art in this way [...] you become alert to subtler nuances of meaning. (2005, p. 111).*

The suggestion of learning the 'language' is a good one, but would not be accepted at face value and undertaken by all. So in current standing, it is fair to say that there is a divide between those with and those lacking the cultural capital to make these judgements. We can conclude that laymen primarily are in contact with 'functional' software, built as a means to an end, whereas experts may interact with software built 'for software's sake', and have a deeper understanding of history and the cultural sphere, allowing them to 'appreciate' art: "A work of art has meaning and interest only for someone who possesses the cultural competence, that is, the code into which it is encoded" (Bordieu, 1984, p. 2). Furthermore, in the case of conceptual software art, to take Bordieu's 'code' statement literally is to ask: is it necessary to understand the programming language to fully appreciate a work of software art? The answer is to some extent, yes - as a classically trained musician can experience a symphony in a vicarious capacity, or as a fine art painter can understand the reasons for certain brushstrokes. Though, what is more important

for the appreciation of art is to hold an understanding of the key concepts and discourses relevant to an artwork, which is something completely independent of being a technician of the medium - the practical application of this understanding of cultural history, or context, is how we bridge the gulf between definitions of 'technician' and 'artist'.

An artist does not exist without a context: It is important to give some thought to the role of cultural institutions. In the case of 'pure' software art (code art), much of the content produced doesn't fall under the remit of institutions such as galleries, and this has political significance, as well as the aesthetic considerations discussed in the previous chapter. We can return to runme.org's definition, that software art "on the one hand brings software culture into the art field, but on the other hand it extends art beyond institutions" (runme.org, n.d). In 'Culture at the Crossroads', Marc Pachter and Charles Landry address the changing cultural environment at the start of the 21st century. On how the role of the artist has changed, they recall that "it would be artists who often ran institutions such as theatres and galleries" (2001, p. 98). In more recent times, "as the management ethos has taken hold [...] a different type of institutional leader has emerged whose task is often to promote and market the economically sound. The artist's role has diminished at the expense of visionary skills" (p. 99). They are describing a saturation of the art world of non-art professionals, a division of labour perhaps. To generalise, this has changed the artists role from being more 'active' and based in community to being less in the public eye, a mystery to the general public, with the air of 'enigmatic creator'. So if cultural institutions, knowledge hegemonies and power structures are all inadvertently alienating the general public and discouraging education and engagement with contemporary art and artists, then this leaves artists with quite a task to engage audiences. To avoid isolating

themselves completely, artists may have to reach their audience where they are. Bourriaud's concept of Relational Art could be considered an optimistic look on how to break down the social barrier erected by Bordieu. Relational Art, the term under which some software art can be included, doesn't "[assert] an independent and private symbolic space" (Bourriaud, 2002, p. 14) because it focuses on human-to-human interaction, albeit in this case, through a medium of a computer. The 'private symbolic space' is one component of the knowledge hegemonies of art and is precisely what contemporary art critics like Spalding rally against, so to be lacking in this is to level the playing field for the audience. Though perhaps not private, a symbolic space still exists in software art–we cannot suppose that everybody knows how to interact with the software, or even the hardware. This is not a problem faced by traditional media like paintings, not only because we've been conditioned to certain archetypes, but rather than gradually interacting to uncover new facets and meaning (as is the case with interactive art), you are presented with the art in its entirety up front. At this point we can see that the pursuit of complete transparency in user interfaces leaves us with a dilemma. While hiding the 'true' functions of computers and leading to a mystification, transparency is so necessary for access, access which can lead to users further understanding of the software–this is a double-edged sword and presents software artists with another decision of how accessible to make their artwork.

In his conclusion of 'Making is Connecting', David Gauntlett (2011) attacks this hierarchy of creativity, claiming that the professionals and companies actively run down amateur and home-made media. He states:

> *'Art' itself has also become a professional field of experts and elites, who carefully police the borders of their practice. A significant part of the joy of craft, and online creativity, is of course that it does not rely on hierarchies of experts and elites to be validated. (p. 1)*

Standards like 'open source' can give end-users "a whole new way of doing business, and the possibility of unprecedented shifts in the power structures of the computer industry" (Connell, 2001) because of the transparency of the process and end product. Open source projects may allow laymen some education in software and programming, but it is still largely an autodidactic process. In current standings, is only opened up to an elite, open source amateurs and enthusiasts tend to get culturally relegated to 'geeks' and 'hobbyists', a construction that serves only to further mystify software art. While technology has made communication, education, and the movement of information infinitely more accessible, its application in the art world has thus far served to fundamentally separate the skilled artist-technician and their enchanted audience, reinstating the position of the 'master' in art.

As this chapter illustrates, the relationship between artist and audience is damaged. The level of damage is sustained in part by cultural institutions which have been filtering direct interaction and education between artist and audience. There could be ground gained in the relationship could by resurrecting the age of artist-lead institutions (as remembered by Pachter and Landry), and educate the masses in programming to facilitate and nurture a society of producers, rather than consumers of software. The internet affords this opportunity and can challenge existing knowledge hegemonies through education, but it will be a struggle: Currently the process isn't intrinsically didactic and the successes and results of the education are not immediate, they need development and take time.

Conclusion

As this thesis has shown, the crossover between technology and art has resulted in fundamental discourses about not only what we can consider to be art, but how we understand ourselves as human beings and relate to computers. Software art has spent decades exploring its own definition through the work of its practitioners, and it still remains a broad category, a vessel for sub-genres and individual artworks. This detail means perhaps the simply the fact of the medium alone cannot inherently provide new aesthetic experiences. What may be catalysts for new aesthetic experiences are the *characteristics* of individual examples of software art, as they may be part of a wave of wider social changes towards the "information age of post-industrial society" (Shanken, 2002, p. 434). These characteristics have been, and will be, providing new aesthetic experiences for audiences as technology develops and culture evolves. Furthermore, if we can successfully apply Clare Bishop's approach of Bourriaud's Relational Aesthetics in practice, we may be able to engineer new aesthetic experiences, providing we can make good use of the *quality* of the relational experience. However the relationship and quality of interaction between artist and audience, as social archetypes, has gradually deteriorated. This degradation is maintained in part, purposefully or accidentally, by cultural institutions that filter direct interaction and education between artist and audience while presenting 'inaccessible' art and internalising its discourse. The breadth of software art means on one hand it serves this 'traditional aura' to gallery audiences, and on the other is a more public-facing operation with a disseminated distribution method. Unfortunately both of these practices are being mystified. Commercial providers of software and technology are playing something of an antagonistic role here. From them we get simple, beautiful machines and transparent user interfaces, which leads to enchantment: This is not

educational, inclusive or critical, but rather leads to a 'sit-back-and-be-told' culture. Despite the best intentions of 'Web 2.0' the processes of our interactions as a consumer of software get dictated to us. Enchantment can't exist with the knowledge of the workings of the interactive system, so hypothetically, in a world full of experts it would be harder to create enchanting experiences. Having this knowledge is a risk we should be willing to take, as it will release us from the ignorance of simply consuming content: This is what we risk becoming, if we continue on the current path. Technology has assimilated many of our social processes. In an imagined (or future) world where everyone is educated on the workings of software, then individuals can understand the processes undertaken by the ubiquitous technology surrounding them and perhaps have the mental facility to situate the humans in relation to computers. It is somewhat paradoxical that with understanding of how computers work we may be able to 'humanise' ourselves and avoid technological reification, or falling for the illusion of reality that the virtual presents us with every day. To some extent the internet is a proponent of virtual reification, with its social networks and virtual worlds, but can also provide the platform challenge this through education. Without this, the knowledge hegemony will remain, and technology will do just as Bourriaud and Adorno feared: "Set up more and more sophisticated subjugation techniques" (Bourriaud, p. 12), unless we can properly harness and guide technology to make true its "promise to liberate people from ignorance" (Zuidervaart, 2003).

Bibliography

Benjamin, W., (1935).*The Work of Art in the Age of Mechanical Reproduction.* [online] Available from: <design.wishiewashie.com/HT5/WalterBenjaminTheWorkofArt.pdf> [Accessed on 6/12/11]

Berger, J., (1972) *Ways of Seeing.* London: BBC/Penguin

Bishop, C., (2004). *Antagonism and Relational Aesthetics*. In *October 110*, October Magazine Ltd, Cambridge, MA: MIT Press

Bolter, J. D., Gromala, D., (2003). *Windows and Mirrors: Interaction Design, Digital Art, and the Myth of Transparency*. Cambridge, MA: MIT Press

Bordieu, P., (1984). *Distinction: A Social Critique of the Judgement of Taste*. [online] Available from: <web.mit.edu/allanmc/www/bourdieu1.pdf> [Accessed on]

Bourriaud, N., (2002). *Relational Aesthetics.* Dijon: Les Presses du réel

Broeckmann, A., (2006). *Software Art Aesthetics*. [online] Available from: <http://www.mikro.in-berlin.de/wiki/tiki-index.php?page=Software+Art> [Accessed 23/11/11]

Burnham, J., (1970). *Alice's Head*. In: Alberro, A. and Stimson, B. (eds.) (1999) *Conceptual art: a critical anthology.* Massachusetts: MIT [online] Available from: <http://emc.elte.hu/seregit/ConceptualArt.pdf> [Accessed on 23/11/11]

Connell, C., (2001) *Open Source Projects Manage Themselves? Dream On.* [online] <http://www.chc-3.com/pub/manage_themselves.htm> [Accessed on]

Cox, G., McLean, A., Ward, A., (2004) *Coding Praxis: Reconsidering the Aesthetics of Code.* In *read_me Software Art & Cultures Edition 2004* [online] Available from <http://art.runme.org/1107862958-4046-0/cox.pdf> [Accessed on 12/1/12]

Dewey, J., (1980). *Art as Experience.* 23rd ed. New York: Perigee

Gauntlett, D., (2011). *Conclusion.* Extract from *Making is Connecting: The social meaning of creativity, from DIY and knitting to YouTube and Web 2.0*. London: Polity Press. www.makingisconnecting.org
Work made available under a Creative Commons Attribution-Noncommercial-Share Alike license.

Gere, C., (2008). *Digital Culture.* 2nd ed. London: Reaktion

Hof, R. D., (2006). *My Virtual Life (Cover Story).* Businessweek, 1 May [online] Available from: <http://www.businessweek.com/magazine/content/06_18/b3982001.htm> [Accessed 5/1/2012]

Lawrence, A., (2000)*The Spatial 'Aura' of Mariko Mori's Pure Land*. In Eskilson, S., Petersen, R., (eds.) *Modernity: Critiques of Visual Culture.* Eastern Illinois University. Available from: <http://www.eiu.edu/~modernity/alawrence.html> [Accessed 5/1/2012]

Livingstone, I., (2012). *Teach children how to write computer programs.* Guardian, 11 January [online]
Available from: <http://www.guardian.co.uk/commentisfree/2012/jan/11/teach-children-computer-programmes> [Accessed 12/01/12]

McCarthy, J., Wright, P., Wallace, J., Dearden, A., (2005). *The experience of enchantment in human-computer interaction.* In *Personal and Ubiquitous Computing Magazine September 2006.*

Murray, J., (2012). *Pupils need to understand computers, not just how to use them.* Guardian, 9 January [online]
Available from: <http://www.guardian.co.uk/education/2012/jan/09/computer-studies-in-schools?intcmp=239> [Accessed 12/01/12]

Pachter, M., Landry, C., (2001). *Culture at the Crossroads*. Bournes Green: Comedia

Perniola, M., (2004). *Art and its Shadow*. London: Continuum

runme.org, (n.d). *About.*
Available from<http://runme.org/about.tt2> [Accessed 23/11/11]

Shanken, E. A., (2001). *Art in the Information Age: Technology and Conceptual Art*. LEONARDO, Vol 35. [online]
Available from: <http://leonardo.info/isast/articles/shanken.pdf> [Accessed on]

Spalding, J., (2003). *The Eclipse of Art: Tackling the Crisis in Art Today*. London: Prestel

Walker, J., (1988). *Through the Looking Glass: Beyond 'User Interfaces'*. [online]
Available from: <http://www.fourmilab.ch/autofile/www/chapter2_69.html>
[Accessed on 5/1/12]

Walsh, P., (2004). *That Withered Paradigm: The Web, the Expert, and the Information Hegemony.* [online]
Available from: <http://web.mit.edu/comm-forum/papers/walsh.html>
[Accessed on 23/11/11]

Zuidervaart, L., (2011). *Theodor W. Adorno*. In Zalta, E. N. (ed.), *The Stanford Encyclopedia of Philosophy Winter 2011 Edition* [online]
Available from: <http://plato.stanford.edu/archives/win2011/entries/adorno>
[Accessed on 16/11/11]

REFERENCED SOFTWARE ART AVAILABLE ONLINE

JODI.org, (n.d), *JODI.org*
Available from: <wwwwww.jodi.org> [Accessed on 22/11/11]

Snibe, Scott., (2002). *Tripolar*, source code from CODeDOC.
Available from: <http://artport.whitney.org/commissions/codedoc/Snibbe/Tripolar_java_wc.html> [Accessed on 6/1/12]