



# *The University of Bergen*

*Department of Linguistic, Literary and Aesthetic Studies*

DIKULT350

Master's Thesis in Digital Culture

Fall 2016

## **HERITAGE.exe**

*The current state and future of Norwegian software preservation —  
mapping the challenges of preserving locally executable digital arti-  
facts and the strategies needed to overcome them.*

Thor Baukhol Madsen

# Abstract

A digital culture is expressed through digital media. This means that a part of our cultural heritage is now embodied in digital artifacts — including software. Using this as a base premise, this thesis aims to disclose and discuss the current state and future of Norwegian software preservation. In order to bring about such a discussion, it maps the challenges related to software preservation and evaluates the currently available strategies for ensuring long-term access to locally executable digital artifacts.

The challenges of preserving software are discussed on the basis of a broad theoretical framework and a case study of three U.S institutions which have all implemented different frameworks for ensuring executable artifacts — namely the Internet Archive, Stanford University Libraries and the Media Archaeology Lab.

- - - - -

En digital kultur uttrykkes gjennom digital medier. Dette betyr at deler av vår kulturarv nå er innlemmet i digitale artefakter — også software. Med dette som grunnpremiss, har denne oppgaven som sin hensikt å avdekke og diskutere den nåværende situasjonen for norsk software-bevaring. For å kunne gjøre dette, kartlegger denne oppgaven utfordringene ved å bevare software og evaluerer de tilgjengelige strategiene for langtidsbevaring av lokalt kjørbare digitale artefakter.

Utfordringene ved å bevare software diskuteres utfra et bredt teoretisk rammeverk samt en kasusstudie av tre ulike amerikanske institusjoner som har implementert tre ulike strategier for å bevare kjørbare digital artefakter — henholdsvis the Internet Archive, Stanford University Libraries og the Media Archaeology Lab.

# Acknowledgements

It took a village to write this thesis. I have met a lot of remarkable people who have helped and guided me along the way. Firstly, I want to express my thanks to the institutions analyzed in this thesis — the Internet Archive, Stanford University Libraries, the Media Archaeology Lab, Nasjonalbiblioteket and Riksarkivet. It would not have been possible to conduct my study without their support.

I would also like to express a personal **thank you** to;

**Scott Rettberg**, for your valuable input, guidance and inspiration throughout my project. Thank you for our on point discussions and healthy digressions.

**Jason Scott**, for having interest in my project and for taking the time to talk with me about the Internet Archive's efforts to preserve software.

**Peter Chan**, for welcoming me at the Cecil H. Green Library at Stanford and for our long discussions on Stanford University Libraries' work on software preservation.

**Lori Emerson**, for showing me the Media Archaeology Lab and for our great conversation on hardware and software preservation.

**Erik Gunnar Aaberg**, for your enthusiasm and help in disclosing the current state of Norwegian software preservation, and for clarifying where Riksarkivet stand on these issues.

**Kjersti Rustad**, for your support and for taking the time to answer my questions about Nasjonalbiblioteket's efforts to preserve software.

**Mark Graham** and **Brewster Kahle**, for welcoming me at the Internet Archive's headquarters and for giving me a tour around the premises.

**Aud Gjersdal**, for your interest in my project and for taking the time to proofread my thesis.

**Meltzerfondet**, for making my travels to the Internet Archive, Stanford University Libraries and the Media Archaeology Lab possible.

**Marianne, family and friends**, for keeping me sane and remind me to have a beer once in a while.

**My fellow students at 304c**, for making this year far better than projected.

# Table of contents

<b>Abstract</b>	<b>ii</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>Table of contents</b>	<b>iv</b>
<b>Introduction</b>	<b>1</b>
<i>Methodology</i>	3
<i>Structure of the thesis</i>	5
<b>1. The challenges of preserving executable digital artifacts</b>	<b>6</b>
1.1 <i>Technical challenges</i>	6
1.1.1 Refuting the myth of immateriality	7
1.1.2 Digital materialist theories	15
1.1.3 Lost in translation — human versus machine language and the problems it poses for digital preservation	19
1.2 <i>Societal Challenges</i>	23
1.2.1 Digital obsolescence	23
1.2.2 Legal issues	26
1.2.3 Recognition	29
<b>2. The whys, whats and hows of software preservation</b>	<b>33</b>
2.1 <i>Why preserve software</i>	33
2.1.1 Software as artistic media of expression	36
2.1.2 Software as historical resources	38
2.1.3 Software as a prerequisite to the preservation of other digital artifacts	40
2.1.4 The imperative to preserve software	41
2.2 <i>What aspect of an executable artifact should we preserve</i>	42
2.2.1 Two levels of authorship — authored code versus intended function	44
2.2.2 Authorship and authenticity, and their impact on software preservation	46
2.3 <i>How to preserve software</i>	48
2.3.1 Hardware preservation	49
2.3.2 Migration	50
2.3.3 Emulation and reconstruction	51
2.3.4 Metadata, paratexts and documentation	52
2.3.5 Finding a viable strategy for software preservation	54
<b>3. Empirical analysis of U.S frameworks for software preservation</b>	<b>55</b>
3.1 <i>The Media Archaeology Lab</i>	55
3.1.1 What is the MAL	55
3.1.2 Challenges	57
3.1.3 The importance and benefits of The Media Archaeology Lab	59
3.2 <i>The Born-Digital Program/Special Collections at Stanford University Libraries</i>	61
3.2.1 What is the Born-Digital Program	61
3.2.2 SUL’s framework for software preservation	63
3.2.3 Challenges and limitations	65

3.3 <i>The Internet Archive</i>	69
3.3.1 What is the Internet Archive	69
3.3.2 Internet Archive's framework for preserving software	70
3.3.3 Challenges and limitations	73
3.4 <i>Comparing the institutions</i>	78
3.4.1 Frameworks	78
3.4.2 Common challenges	80
<b>4. Current situation for software preservation in Norway</b>	<b>81</b>
4.1 <i>Historical outline of digital preservation efforts in Norway</i>	81
4.2 <i>The status quo and future of Norwegian software preservation</i>	84
4.2.1 Responsibilities and implementation	84
4.2.2 Copyright and limitations on access and use	86
4.2.3 Discussion – strengths and weaknesses	88
<b>5. Conclusion and ending thoughts</b>	<b>95</b>
5.1 <i>Threats, challenges and motivations</i>	95
5.2 <i>A viable strategy for software preservation</i>	97
5.3 <i>Norwegian software preservation</i>	98
5.3.1 Findings and issues	98
5.3.2 Future research	102
<b>Bibliography</b>	<b>103</b>
<i>Interviews/personal correspondence</i>	103
<i>Literature</i>	103
<b>Appendix</b>	<b>110</b>
<i>Interview questions</i>	110
Stanford University Libraries and the Internet Archive	110
The Media Archaeology Lab	110
Nasjonalbiblioteket and Riksarkivet	111

# Introduction

A digital artifact is a conditional entity of information that cannot exist independently in the world as a humanly meaningful object. Or more precisely, a digital artifact cannot exist as such outside the confines of a digital computer system or storage device. However, the ease of which a digital entity of information can transgress the boundary between the digital and the non-digital realm rely on its complexity. Remediations of print technologies, like a novel written in a word processor, or a drawing made in a graphics editor, can often be printed to paper without any loss of information or meaning. This is not the case when it comes to executable digital artifacts. In contrast to a linear text or a static image, software can offer ever-changing and interactive environments that cannot be divorced from the digital realm through acts of solidification, without losing some of its properties or initial meaning. If we print all the pages of a hypertextual work or every still image from a computer game, what we end up with is a documentation of the work rather than a copy of the work in its entirety. The work is thus transformed into a static digital object and deprived of its interactivity, dynamics, randomness and performativity. This poses new challenges for memory institutions who wish to preserve software, in the sense that these artifacts cannot be separated from the digital realm without losing meaning or changing its intrinsic properties.

Within a digital culture, an increasing amount of our cultural heritage is embedded in digital artifacts. Some of these artifacts exist in software form, or they depend on some kind of software intermediary to be translated into humanly meaningful information. Consequently, a digital culture is also expressed through the software and software-dependent artifacts it produces. In this regard, the premise of this thesis is that software are important cultural and historical artifacts whose materiality brings new challenges for the memory institutions who are trying to preserve them.

The technical challenges of preserving executable artifacts, I argue, are further amplified by the societal factors of obsolescence, law and status. While obsolescence reduces the lifespan of digital materials, the legal framework may directly or indirectly limit archives' ability to preserve them. Finally, the

preservation of executable digital artifacts depends on them being recognized by archives, and society in general, as valuable cultural artifacts.

This thesis offers a pragmatic approach to software preservation — it discusses the technical and societal challenges of preserving executables and evaluates different strategies for overcoming these problems. Currently, the scholarly approaches to software preservation appears to be divided into two camps; 1) a media archaeological approach, in which digital artifacts are preserved by sustaining the obsolete computer systems needed to run them, and 2) a digital archivist approach, in which the information is preserved separately from its original hardware and software environment. While the former resembles the traditional methods of archiving in the sense that it archives information by preserving the media onto which it is stored, the latter preserves it through transcription — by copying the information from the original media and onto new and safer storage systems.

In order to evaluate these two main strategies for software preservation, this thesis offers a comparative case-study of three different U.S memory institutions that have each developed different frameworks for preserving software. The case-study analyses the frameworks of the Internet Archive, Stanford University Libraries and the Media Archaeology Lab, and maps the challenges they have encountered along the way. While the MAL preserves executables indirectly by sustaining obsolete computer systems, the IA and SUL preserves software by separating it from its original environment.

Finally, this thesis aims to disclose the current state of Norwegian software preservation and to determine whether any of the nation's prime memory institutions have a framework for preserving these materials in place. Using the comparative case-study as a basis, I further discuss the current Norwegian frameworks and discuss whether or not these are sufficient for ensuring long-term preservation of, and access to, the nation's software-based heritage.

The research questions of this thesis can be summarized as follows;

1. What are the challenges of preserving executable digital artifacts, and why should we bother to overcome them?
2. How do we preserve executable artifacts, and what methods constitute a viable strategy for the long-term preservation of software?

3. Do Norwegian memory institutions have a viable strategy in place, and if so, how does it compare to the frameworks of the U.S institutions? Does it sufficiently ensure long-term access to Norway's software-based heritage? What is the status quo of Norwegian software preservation?

## Methodology

The first half of this thesis discusses a broad theoretical framework that incorporates literature from the field of humanities together with legal and archival theory. However, as software preservation is a relatively new discipline, there was a limited amount of research materials available. To properly map the challenges and strategies related to software preservation, it was therefore necessary to make contact with institutions that are actively preserving software. In order to thoroughly examine different approaches to software preservation, I decided to do a comparative case study of three U.S institutions that are using separate methods to actively preserve software. This allows for a deeper understanding of how institutions are dealing with these artifacts in practice, and it opens for a more detailed mapping of the challenges involved in ensuring long-term access to software. These institutions may, for instance, have encountered different kinds of issues, or they may have run into problems that are not described in the literature. In this regard, a comparative case-study can offer insight into the praxis of software preservation and offer access to information that is not available elsewhere.

The comparative case study analyses and compares the practices of three institutions, namely the Internet Archive, Stanford University Libraries and The Media Archaeology Lab at the University of Colorado. The main reasons for choosing these specific institutions are threefold. Firstly, they are all renowned institutions that preserve software. Secondly, their frameworks incorporate different methods for preserving executable artifacts, and thirdly; together they cover the three main strategies described in archival literature, namely migration, emulation and hardware preservation.

In order to conduct the case-study, I spent around a day at each institution, trying to learn more about how they operate and how they are dealing with the



challenges related to software preservation. I interviewed Jason Scott at the Internet Archive, Peter Chan at Stanford University Libraries and Lori Emerson at the Media Archaeology lab. Lori Emerson is a key figure in the scholarly fields of media archaeology and media studies, and she is the founder of the Media Archaeology Lab. Currently, the MAL holds a hardware collection of 148 obsolete machines, together with large amounts of software that depends on these machines to execute. Further, Jason Scott is in charge of the Internet Archive's software preservation efforts and is the initiator of the Internet Arcade project, which is an emulated arcade game collection that can be accessed and played through the archive's website. Finally, Peter Chan is a digital archivist at the Born-Digital Program at Stanford University Libraries, under the department of Special Collections. They currently own and preserve two of the world's largest software collections, namely the Steve Meretzky and the Stephen M. Cabrinety collections.

Due to the limited amount of previous research within the field, it was important to have a pragmatic approach to these institutions and ask as open questions as possible. The conducted interviews were therefore conversational and informal in their nature. To allow for this, I sent the interviewees questions and themes for discussion prior to our meetings and tried to interrupt them as little as possible during the actual interviews. The transcripts of the audio-recorded interviews are quoted throughout the thesis, and the quotes were confirmed with the interviewees before it was submitted. In conclusion, the comparative case study lays the ground for understanding how the different methods for software preservation are applied and how they work in praxis, and it gives detailed insight into the challenges that these institutions have encountered along the way. These findings are further used as a basis for discussing the strengths and weaknesses of Norwegian software preservation.

With the exception of legal documents and white papers, there are little to no research or literature describing the current state of Norwegian software preservation. To examine this field, I conducted e-mail interviews with Nasjonalbiblioteket (The National Library of Norway) and Riksarkivet (The National Archive of Norway). My interviewees were section manager Kjersti Rustad at Nasjonalbiblioteket, and Erik Gunnar Aaberg, senior advisor at Riksarkivet's digital depot. The reasons why I chose these two institutions are twofold. Firstly, these are the two largest memory institutions in Norway that

actively preserves digital materials, and secondly, these are state institutions that share the responsibility of ensuring the nation's historical and cultural heritage and can thus be held accountable if access to these materials is lost.

The data collected through interviews and e-mail correspondence fills the gaps created by the lack of pre-existing literature and provide the basis for answering the research questions of this thesis. Also, they allow for further discussions on the future of Norwegian software preservation, i.e. what challenges they can expect to meet and how they can avoid some of the issues encountered by the U.S institutions. Finally, they provide the means to assess what constitutes a sufficient strategy for ensuring software-based heritage. Thus they can disclose the strengths and weaknesses of the potential Norwegian strategies, and give clues as to which elements such a strategy should include.

## Structure of the thesis

Chapter one offers an outline of the technical and societal challenges of preserving software. I start by discussing the materiality of digital artifacts, drawing on a theoretical framework of digital-immaterialist and materialist theories. I claim that the immaterialist perspective misses the fundamentally physical nature of digital artifacts and thus is not a viable starting point for building a strategy for software preservation. Based on digital materialist theory, I argue that digital executables are performative and multilayered artifacts that are always traceable to some physical states within the components of the computer. Thus the technical challenges of preserving these artifacts are not grounded in any immaterial properties intrinsic to digital artifacts. Instead, the challenges are rooted in the dichotomy between machine and human language, and the dependency on having the appropriate hardware and software environment needed to translate digital artifacts into humanly meaningful information. In turn, this dependency might make digital artifacts susceptible to societal factors such as obsolescence. The societal challenges are further mapped using a theoretical framework of legal, economic and archival literature.

Chapter two begins with a discussion on why we should preserve software. I argue that executable digital artifacts are part of a digital culture's heritage, and I

offer different reasons for preserving software based upon literature from a wide range of academic fields, including cultural studies, media studies, game studies, archival theory and art history. Further, I discuss what aspects of a piece of software should be preserved. In contrast to analog cultural artifacts, I argue, there are two levels of authorship to executable artifacts — the code and the presentation produced on the basis of that code. Lastly, I introduce the different methods that are currently being used to preserve software.

Chapter three offers a comparative case study of three real-world frameworks for software preservation. This section further examines the methods outlined in chapter two and gives a detailed description of the challenges encountered by these institutions. Finally, chapter four discloses the current state of Norwegian software preservation and offer a discussion on the weaknesses and strengths of the current efforts to preserve executable digital artifacts.

# 1. The challenges of preserving executable digital artifacts

## 1.1 Technical challenges

The complex materiality of digital artifacts separates them from those that are directly available to us as worldly and unified objects. Digital information is encrypted in binary form, as a set of physical but microscopic states within the components of the computer. While the information of a book is inscribed onto paper using ink, digital information is inscribed onto a wide range of materials, such as silicone, metals and plastic, with the help of lasers, electromagnets and electricity. Although digital artifacts have been removed from the channels of direct human intervention and placed outside of our peripheral vision, digital data can still be traced back to physical states and inscriptions. (Kirschenbaum 2008, 86; Kittler 1997, 147–48) If we investigate the platter of an optical hard drive with a microscope, we would see the bits as physical inscriptions measuring about 4 x .5

microns — each of them similar, but none of them exactly alike. (Kirschenbaum 2008, 61–62)

In this chapter, I argue that the issues of preserving digital artifacts are not related to a lack of materiality. Rather, these issues are deeply rooted in the encrypted and performative nature of digital technologies, and in the semantic gap between machine language and human language. What we see and interact with on the screen does not mean anything to the computer. A piece of digital art is not a piece of art to the computer — it is just a series of bits that can be computed and transmitted to a presentational device as an organized collection of pixels. In contrast to conscious beings, the computer does not possess intentionality. (Maslin 2007, 146) On the other hand, the bits themselves do not mean anything to humans. Thus, in order to have access to digitally stored data, we depend on a computer intermediary to translate the bits and present them as humanly meaningful information. This dependency, I argue, is what lies at the core of the technical challenges concerning digital preservation — not immateriality.

### *1.1.1 Refuting the myth of immateriality*

In popular media, the digital realm tends to be characterized as antonymous to the physical realm. In an article published by *The Guardian* in 2013, the headline is as follows; “Forget digital media; let’s get physical”. (Andrews 2013) Drawing a line between a physical world of atoms and a non-physical world of bits, the author argues that he senses a counter-movement to the digital revolution, driven by a “yearning for tangible engagement that can only be achieved through the tactile experiences digital content have eradicated.” (Ibid.) There are several other interesting remarks in this article, purveying the author’s underlying views on the materiality of digital technologies. For instance, he states that “When the internet crept slowly into our consciousness in the middle of the 1990s, its world of bits delighted us with its promise to free us from the tyranny of atoms.”, and he remarks that “..in a world dominated by bits, atoms are due a comeback..” (Ibid.) He appears to build his argument on the premise that bits and atoms are dichotomous entities on the basis of the former’s lack of material properties.

The idea that the digital information lacks materiality, or that it can secede from the material world, also appears in the Norwegian press. In a newspaper article published by *Verdens Gang*, a music critic states that although music is an art form that can be rendered digitally and be liberated from objects, some albums have certain qualities that make it desirable for us to own them in a physical tangible format. (Aanstad 2011) If his statement is taken literally, he seems to imply that digital data exist as pure information, independent of any material underpinnings. In an article, published by *The Wall Street Journal* in 2014, the wording is a bit different. Referring to the “Internet of Things”, the authors assert that “the digital and the physical realms are merging at an accelerating pace.” (Rosenbush and Norton 2014) Underlying this statement is the notion that they were separated to begin with. That they are disconnected worlds that are starting to connect with each other, as we start embedding sensors and wifi chips in everyday objects. Similarly, in a piece by *The Guardian*, it is claimed that “As boundaries blur between the digital and the physical, artists are remaking the metropolis into a playful, human experience.” (McMullan 2015)

One cannot, of course, draw any large-scale conclusions about popular media’s views on digital materiality on account of the few examples mentioned. However, the wording leaves cues for further discussion. Is the digital world antithetical to the physical world? Is the word physical used as a way to assert the intangible aspect of digital artifacts, or does it involve a categorical rejection of their physical properties? In terms of the two first articles, the latter interpretation seems to underlie some of the statements they are making. Consider, for instance, the statement; “.in a world dominated by bits, atoms are due a comeback..”. (Andrews 2013) Is the digital artifact not comprised of atoms? If not, then what is it? Or perhaps the author uses the term ‘physical’ loosely, so as to denote the intangible and seemingly invisible nature of digital artifacts? On the one hand, Andrews separates between atoms and bits —physicality and non-physicality— while on the other, he describes the digital as intangible and invisible. A lack of tangibility or visibility does not however necessarily entail a lack of physicality. Magnetism and electricity, for instance, can both be traced to the elementary particles such as the electron — a subatomic particle which contains mass and is fundamental to matter. (F. N. H. Robinson 2016a; F. N. H. Robinson 2016b; “Electron” 2016)

Jean-François Blanchette (2011, 5) states that “the trope of immateriality is more than a convenient metaphor information age pundits reach for to cut through technical complexity” and that it “plays a central role in several important arguments over the implications of our current society-wide shift to digital information.” We find evidence for such a claim in the writings of academics such as Nicholas Negroponte and Paul M. Leonardi.

In his book, *Being Digital*, Negroponte (1995, 11) separates between the material world of atoms and the digital world of bits. He regards bits as colorless, sizeless and weightless states of being — that is, as binary states of on or off, true or false and so on. (Ibid., 14) The premise on which he bases his argument is that bits are units of pure information, standing ready to serve whatever purpose we deem fit. For instance, a broadcaster could transmit a unified bitstream of a football game that could later be converted into many different formats on the receiving end. We could choose whether to experience the game as a television or radio broadcast, or as graphical data of the scores and plays. All derived from the same stream of bits. (Ibid., 71-72) As a consequence, Negroponte concludes that “the medium is not the message in the digital world. It is an embodiment of it.” (Ibid.,71)

Following this line of thought, he seems to be treating bits as purely logical and conceptual units of information. As weight- and sizeless states of being, freed from the material world. The deductive reasoning leading to such a conclusion can perhaps be construed as follows; 1) bits are entities of information, 2) a piece of information is an immaterial entity, 3) hence, bits are immaterial entities. Nevertheless, for this to be true, the premises need also be true. If bits really are weight- and sizeless states of being, i.e. pure immaterial information, they should be untraceable in the material world. However, if we reconsider Kirschenbaum’s evidence for the traceability of bits —namely that bits can be traced back to physical inscriptions— then Negroponte’s claim simply cannot be true. A closer investigation of the workings of digital computers and storage media further refutes the idea of immateriality.

Essentially, a computer is a collection of on and off switches that produce binary information in the form of 1s and 0s. (White 2008, 49) These binary digits are often referred to as bits. In modern computers based on microchip technology,

the two integers are normally produced and processed by microscopic switches called transistors, which control the flow of electricity running through them. If the current is allowed through, the transistor creates a 1, if the current is stopped, it creates a 0. (Ibid., 53) By controlling the flow of electricity, the transistors can not only record and manipulate numbers, but they can also deal with Boolean logic by assigning truth values to the binary information so that 1 stands for true and 0 for false. (Ibid., 51, 53) In this way, the computer can use bits, 1s and 0s, to manipulate and store all of its information. (Ibid., 57) A functional computer is in this regard constituted by physical phenomena and objects. Even at the computational level, bits are not weight- and sizeless states of being, they can be traced to specific states in a set of physical switches. Although microscopic in scale, transistors are objects that perform calculations by manipulating electricity. Furthermore, electric charge is born by the electron particle — a particle which has mass. (“Electron” 2016)

Bits are also traceable at the level of storage. Although there are many different kinds of storage media, the data is stored either by carefully arranging particles, controlling states of electric charge or by physically manipulating some kind of surface material. When storing on a disk drive, the bits are inscribed by organizing iron particles into data using an electromagnet. (White 2008, 158) It does this by creating two bands of iron particles, which are either magnetized in the opposite or the same direction. If the particles are magnetized so that the poles of the two bands face in the same direction, they make a 0, if they are faced in opposite direction of each other, they make a 1. (Ibid., 158–59) If the file is stored on a compact disc, such as a CD-R, a laser is used to inscribe a special code of compressed data. The high-powered beam of light creates marks on the disk that can later be decompressed by the CD-drive and passed on to the computer as 1s and 0s. (Ibid., 189) Flash memory works somewhat differently from other kinds of storage media, as it memorizes data by controlling the electrical charge of billions of cells. Simply put, if a cell has a charge of over fifty percent, it represents a 1, and if not, it represents a 0. (Ibid., 60-61)

Although microscopic in their nature, all elements in play have physical properties, and with the exception of photons, all elements involved have mass. It might be that the idea of immateriality stems from the obscured and complex nature of digital technology and the terminology we apply to it. A word like materiality

brings about connotations of tangibility and visibility, and perhaps this lies at the heart of some claims about immateriality — that the deep mechanisms of the computer are inaccessible to us on a sensory level. We cannot reach out and intentionally touch a single bit, nor can we see it with the naked eye. However, an entity need not be directly sense-perceptible for it to have materiality. If that was the case, a speck of dust or a water molecule would fall within the category of immaterial entities.

Perhaps the dispute between the materialist and the immaterialist position is rooted in how they define *materiality*. Paul M. Leonardi (2010, 2) argues that “if materiality is defined simply as matter,.... digital artifacts cannot be said to have materiality.” He therefore proposes that we turn to other definitions of the term when describing digital artifacts — so as to denote significance or to point to the practical instantiation of a digital artifact. He writes;

A digital technology like a word processing program or a finite element solver is an artifact that is not comprised of matter and cannot be touched or, in the case that it runs in the background on one’s computer, even pointed to. Although it has no physical properties, software clearly does not exist in the conceptual domain because it provides hard constraints and affordances in much the same way as physical artifacts do. (Ibid., 2)

Due to the absence of physical properties, he argues that we cannot apply the term materiality to digital artifacts, as long as it is characterized as follows; “Senses related to physical substance. Formed by or consisting of matter” (Ibid., 4) However, this definition does not seem to be directly incompatible with digital artifacts. Are our sensations of the software not related to physical substances? The software is stored on a hard drive, processed by microchips, presented to us on a screen and we interact with it using a mouse and a keyboard. Leonardi attempts to refute such a counter-argument by saying;

Although critics would be correct that one cannot experience MS Word without a monitor, keyboard or hard drive, the software has no physical



matter of its own (even though its code is mediated by electronic signals and it relies on physical artifacts such as disk drives and processors for its execution. (Ibid., 4)

No, we cannot reach out and grab, or touch, the software. And no, data and electricity are not objects. (Ibid., 2) However, this does not necessarily entail that the data lacks physical properties or materiality. Leonardi is perhaps rather implying that a piece of software exists somewhere outside of the computer, like some sort of Platonic idea, a manifestation that is realized by the computer but not equal to the software itself. This is especially apparent in his proposal to use the word “material” so as to denote the practical instantiation of a theoretical idea. (Ibid., 2) Although this might be a fair point to make in relation to our *experience* of the artifact, it does not entail that the software is immaterial or non-physical in itself. It appears that Leonardi is ascribing digital artifacts with a similar kind of materiality as that which is often ascribed to thoughts or consciousness. However, although a thought appears not to be equal to the measurable activity in the brain, this is not necessarily evidence for the immateriality of thought. It cannot be ruled out that consciousness might be identical to the processes in the brain, such as the measurable blood flows or neural activity, and it cannot be ruled out that conscious entities might be physical arrangements of some sort. (Maslin 2007, 55) The relationship between software and hardware is of course not as abstruse as the relationship between cognition and the brain. The computer is not a conscious being, nor does it possess intentionality.

Perhaps part of the issue is the underlying notion that something can exist in itself. This is not a problem if we use it to account for the totality of the parts involved, such as describing a specific book without having to point to the parts of which it is comprised — i.e. paper, ink, leather and so on. The problem occurs when it is used to point to some kind of form outside the material world. Both Negroponte and Leonardi seems to argue from a substance or property dualist point of view, in which the properties of digital artifacts are associated with the properties of the mind. According to Kenneth J. Knoespel and Jichen Zhu (2008, 235), the tendency to regard computer code as immaterial can be linked to the long tradition of Cartesian dualism and the influence it has had on linguistic theory.

Within such a conceptual framework, the human capacity for language is seen as separate from the material world. Computer code, and information in general, can from such a perspective be considered immaterial.

It seems that Leonardi's position on the relationship between software and hardware is quite similar to how property dualists define the relationship between the mind and the brain. In contrast to substance dualism—in which human beings are considered a composite of two disparate entities, a non-physical mind and a physical body—property dualism maintain that the brain instead possesses two fundamentally different kinds of properties, i.e. physical properties and non-physical mental properties. (Maslin 2007, 30–32) Although Leonardi states that software has no physical properties, he adjusts his claim by adding that it does not exist in the conceptual domain, as it provides hard constraints and affordances in the same way as physical artifacts do. (Leonardi 2010, 2)

However, the argument for digital immaterialism is subject to the same critique as substance and property dualism. Descartes' dualist account of how the immaterial soul controls the body through the pineal gland, sets him in a position of which he would have to explain the causality of how something immaterial could move something of matter. (H. Robinson 2003, sec. 1.2) In Leonardi's case, he would have to explain how the hardware of the computer can produce something that has no physical properties. Further, he would have to account for how this immaterial entity can be converted into something that has physical properties—that is, pixels on a screen, sound through a speaker and so forth. How can something without physical properties have a practical instantiation, and how is software able to cross the boundary between the non-physical and the non-physical? If one conforms to the laws of physics, there seems to be a logical fallacy underlying the idea of digital immaterialism. Consequently, immaterialist views necessitate some kind of explanation of the causal relations between hardware and software, something of which neither Negroponte nor Leonardi provide.

The term materiality contributes perhaps to the perceived elusive and mysterious nature of the digital world. It is difficult to make a functional computer fit within a strict definition of materiality, in terms of it simply being dead matter. The physical components of the computer, such as the hard-drive or the processor, are composed of matter and are tangible and visible objects. However, by

themselves, these components would not be able to compute anything. They depend on natural and physical phenomena such as electromagnetism, electricity, and light to perform calculations. As a consequence, the computer cannot be solely defined on the basis of the hardware constituents that comprises it, as it is contingent on physical and natural phenomena to operate. The entirety of the functional computer can therefore not be reduced to a single confined and dead object, but this does not entail any lack of physicality. The digital artifact exists as the sum of all the physical processes inside the computer. It is a series of bits, but the bits are either inscriptions on a CD, electromagnetically organized iron particles, a state of stored electricity and so on. It is true that a digital artifact is not an object, rather it is the sum of a series of physical processes and states within the components of the computer. It is a composite and performative artifact consisting of multiple layers.

Consider, for instance, a sun setting over the horizon. Due to the rotation of the earth, the sun appears to be slowly engulfed by the ocean, scattering hues of orange and red in the atmosphere. The complex image that hits our eyes is however formed by a series of physical phenomena, matter and liquids — the sun, the earth, the earth's spin around its own axis, atmospheric refraction, water molecules and so forth. If you separate the processes and look at them individually, what you have is not a sunset. The process of which the earth spin around its own axis is not equal to the sunset we perceive, nor is the atmospheric refraction, the ocean or the clouds. It is the interplay between all of these factors that constitutes a sunset. It is performed by a series of elements that need to be set in a specific state in relation to each other, in order to produce what we perceive as a sunset. Furthermore, none of the elements can be said to be non-physical. The qualia that comprises the subjective experience of the sunset might involve non-physical aspects —that is an entirely different and more comprehensive discussion— but the elements that form the output from which the phenomenological experience is made possible, do not. Not all of the components are tangible, visible or have mass, but they are not non-physical concepts. Although light is not a tangible and confined object, ultraviolet radiation can age our skin and cause eye diseases and skin cancer — all of which are physical effects imposed onto our bodies by the physical properties of light.

### 1.1.2 Digital materialist theories

The perceived digital artifact is the product of the inter-relational workings of the physical components in the computer. The idea of immateriality stems perhaps from how the digital artifact is perceived, rather than its actual physical properties. Similarly, Kirschenbaum (2008, 61) differentiates between forensic and formal materiality. While forensic materiality points to the physical traceability of bits, formal materiality is concerned with the symbolic and logical nature of bits and defines the computer as a tool for manipulating symbols. The latter conception underlies many claims about immateriality, including Negroponte's distinction between bits and atoms. (Ibid., 11) The seemingly immaterial nature of digital technologies could be a direct result of a lack of transparency. Kirschenbaum writes;

Whereas forensic materiality rests upon the potential for individualization inherent in matter, a digital environment is an abstract projection supported and sustained by its capacity to propagate the illusion (or call it a working model) of immaterial behavior: identification without ambiguity, transmission without loss, repetition without originality. (Ibid., 11)

In this regard, the immateriality of digital artifacts is an illusion propagated by the very design of the computer. In contrast to traditional writing practices, digital writing conceals the actual inscriptive process and becomes friction free. As a consequence, electronic text is perceived as an “essentially symbolic rather than inscriptive exchange among a set of operational metaphors and the electronic elements on the screen” (Ibid., 41) This is reflected in the language we use to relate to digital inscription. For instance, he argues, it is common to refer to the process of storing a file as a process of writing it *to* a disk, rather than writing it *on* a disk. This preposition contributes to “a sense of inferiority; because we cannot see anything on its surface, the disk is semantically refigured as a volumetric receptacle, a black box with a closed lid” (Ibid., 87)

Perhaps some popular claims of immateriality can be linked to the concept of the ‘black box’ — a term for the computer, used to denote the inaccessibility

and complexity of its inner workings. (Parikka 2015, 2758) For the regular user, a computer is an object that responds to a given input by producing an appropriate output. It is intentionally designed to “render the mechanism invisible and usable as a single functionalized object.” (Ibid., 2778) Similar to the design of a matryoshka doll, it is constituted by layers upon layers of hardware decreasing in size — from the level of perceptible objects to the microscopic and hidden level of particles. Although many know how to use a computer, most do not understand how it works. This lack of insight may very well have fueled the conception that digital artifacts are immaterial in their nature.

Similarly, Jean-Francois Blanchette (2011, 2) asserts that the computing infrastructure is built to relieve users from the constraints of the material resources of computation. For instance, operating systems and networking protocols ensure that applications can run regardless of processor type, storage media or network connection. However, he claims, this is an abstraction that can never fully succeed, as computation is in fact “a mechanical process based on the limited resources or processing power, storage and connectivity.” (Ibid., 2) According to Blanchette (Ibid., 8), information systems can be divided into three major types of components — applications, infrastructure software and computing resources. While applications provide services to users, infrastructure software mediates applications’ access to the shared computing resources — that is, to the hardware that provides processing power, storage and networking abilities. (Ibid., 8) In order to deal with the complexity of the communication between these factors, the hardware and software industry incorporated a modular design strategy — separating an artifact’s functional specification from its implementation. (Ibid., 9) For instance, the modular design strategy allows for connecting a wide range of hardware components to the computer through a single peripheral interface specification, such as USB. The interface specifies both the services the hardware must provide, as well as the software and hardware language needed to interact with the module. (Ibid., 9) Further, modularity involves layering — the process of which modules are stacked in a series of client-server relationships so that the each layer works as a client to the layer below, and as a server to the layer above. In this way, “bits move up from their grounding as signals in some physical media... to binary information organized according to units defined by each layer ...” (Ibid., 9-

10) Applications can access these stacks through APIs (application programming interfaces) to the various modules of the operating system. (Ibid., 10)

Blanchette regards the materiality of digital information as the composition of two different sets of constraints — the limited resources of computation, and the modular design that allow for communication between these resources and the applications that manipulate the information. (Ibid., 23) In addition, he continues, these resources always deal with bits as physical quantities, whether they take the form of electric voltages, magnetic polarities or radio waves. In other words, digital information cannot be processed unless it takes some kind of physical form. (Ibid., 23)

Knoespel and Zhu (2008, 236) suggest that we use the term continuous materiality to describe the materiality of digital artifacts. This points to a wide spectrum of materiality brought forth by a hierarchy of code, which moves from fundamental machine code to the higher levels of readable code, based on formal programming languages. In this regard, the materiality of digital technologies can be characterized as a pyramid of code, in which each level engages the natural and the physical world in a variety of ways. Altogether, “the hierarchy of codes constructs a field of diverse materiality that is continuous and interconnected.” (Ibid., 236) Code can thus be said to have a presence that extends beyond the perceived virtual space presented by the screen. They are “patterns and sequences discovered or invented by humans to provide access to natural phenomena.” (Ibid., 243) Even though they include logical codes, they are also characterized by evolving interactions and chemical chains. (Ibid., 243)

While Kirschenbaum and Blanchette are committed to proving the materiality of digital artifacts by disclosing their physical properties, Johanna Drucker goes a step further. Building on Kirschenbaum’s differentiation between forensic and formal materiality, she argues that we must go beyond the ontological understanding of materiality, and draw our attention to the performative aspect of digital artifacts. (Drucker 2015, para. 4) She asserts that the concept of materiality should be extended to include its performative dimension. As a result, materiality should not only be defined in relation to *what something is* but also in respect to *what it does*. This idea is in opposition to literal materiality, in which material properties are assigned intrinsic value and objects are thought of as self-identical

and observer-independent. (Ibid., para. 16) She claims that the literal approach is not applicable to digital artifacts, as these are neither self-identical nor observer-independent entities, but rather interpretative events provoked by formal, material properties that provide stimulation. In other words, the “material conditions provide an inscriptional base, a score, a point of departure, a provocation, from which a work is produced.” (Ibid., para. 8)

For Drucker (Ibid., para. 12), the performative dimension lies in the user’s ability to perform a digital artifact. She agrees with Kirschenbaum in respect to the traceability and materiality of digital artifacts but adds a phenomenological and semantic level in which the artifact is given meaning. Although an artifact has specific properties, what those properties *mean* are very different from what they *are*. (Ibid., para. 20) Drucker, therefore, separates between what she calls distributed and performative materiality. While the former focuses on the “complex interdependencies on which any digital artifact depends for its basic existence”, the latter “emphasizes the production of a work as an interpretive event.” (Ibid., para. 21) The digital artifact is in this regard both a product of the interrelated activity across multiple systems and components, as well as a product of human engagement, interaction and interpretation. It is not only shaped by its own properties; it is also formed by its encounter with the cognitive capacities of the user. Performative materiality is therefore always probabilistic, as it “demonstrates the fact that material forms are only the site of potential for meaning production, not for transfer.” (Ibid., para 24)

In summary, these materialist theories refute the idea that digital artifacts are immaterial entities unfettered by matter. As argued by Kirschenbaum and Parikka, digital immateriality is an illusion purveyed by the very design of the computer, which intentionally hides its mechanical and physical nature. While Kirschenbaum asserts that bits can always be traced to some kind of physical inscription at the level of storage, Blanchette shows how the bits are also physical quantities at the level of processing. Finally, Drucker points out that although digital artifacts are always traceable to physical substances, we also have to take into account the performative and semantical aspect of digital artifacts.

Drawing on this theoretical framework, digital artifacts seem to be formed by two levels — a non-semantical and a semantical level. While the former points

to the states and processes within the computer, the latter points to its presented output, i.e what we interact with on the screen. While both of these levels are physical, only the latter means something to humans. The mistake of Nicholas Negroponte and Paul M. Leonardi is perhaps that they confuse a lack of semantics with a lack of materiality. Although an instance of Microsoft Word is not a recognizable or meaningful artifact at the level of bits, this does not entail immateriality. Rather, a digital artifact is formed by a continuous translation between formal and natural language. What we interact with on the screen is not equal to the processes inside the computer, but none of these levels are immaterial or non-physical.

The core problem of preserving digital artifacts can thus be attributed to a dichotomy between machine language and natural language, rather than a notion of immateriality. The issue is not that the bits are immaterial, but rather that they are incomprehensible for human beings. Thus, in order to access a digital artifact, we depend on computer intermediary to translate the bits into humanly meaningful information.

### *1.1.3 Lost in translation — human versus machine language and the problems it poses for digital preservation*

If we approach digital technology only through romantic notions of immateriality, of some ethereal half-life or as an isolated in-between state removed or ever at distance from the real world, we will continue to ascribe to a simplistic realism that bogs down in net-cartesian distinctions which create an illusion of separateness. (Knoespel and Zhu 2008, 244)

The notion of digital immateriality is a problematic starting point for any discussion about digital artifacts — even more so, within the context of digital preservation, How can one construct a framework for preserving our bit-based heritage, if digital technology is ascribed with the intrinsic property of immateriality? As in the case of Negroponte (1995, 11), bits become colorless, sizeless and weightless states of being. If we extrapolate on such a view, the



properties of the digital world seem more closely related to the properties of the mind than to the corporeal — he treats digital objects as purely logical objects, somewhat separated from their actual dimensions in time and space.

Imagine that we are constructing a framework for preserving software and that our point of departure involves a characterization of the software at hand as a sizeless and weightless entity. What we are left with is a digital object that is purely logical — a unit of pure information. However, the rendition of bits as pure information, as something that exists in some kind of abstract non-physical space, may lead us down the treacherous path of ascribing bits with the property of immortality. This is evident in the assertions of George Paul (2009, 19) when he claims that digital writing does not depend on the alteration of matter, and should be considered as something that is very close to pure information. In writing today, he states, “we deal in pure information objects, unfettered by matter.” (Ibid., 19) However, if bits are unfettered by matter, bits should not be vulnerable to material decay.

Nevertheless, both Negroponte and Paul are correct in asserting the logical and informational dimension of digital objects. The problem, however, arises when the logical and the material aspects of digital artifacts are dichotomized. The presence of logical properties does not exclude materiality and vice versa. Take a mechanical watch, for instance; when energy is applied, it moves several precision-cut gears at a specific pace, allowing for an accurate representation of time. After a second, the second hand moves, after a minute, the minute hand moves and after an hour, the hour hand moves. The mechanical watch adheres to some set rules that are incorporated in the very parts of the machine, a logical scheme that allows it to express information. The logical aspect of a mechanical watch is not separated from its mechanical function; it is produced by the material parts that comprises it. Similarly, the logical aspect of a digital computer, although far more complex in its workings, is a subsequent event of the rule-bound manipulation of its parts through the suppliance of energy. In relation to digital preservation, one must deal with a digital artifact both as a logical and a material unit, and recognize that these aspects are intertwined — not a matter of either-or. If the materiality of digital artifacts is overlooked, we may fail to acknowledge the vulnerabilities that it causes.

In his article, “Overview of Technological Approaches to Digital Preservation and Challenges in Coming Years”, Kenneth Thibodeau recognizes this. Although separating between the terms physical and logical, he demonstrates the multifaceted and composite materiality of digital entities. The materiality, he proposes, is threefold — the digital object is a physical, logical and a conceptual object. (Thibodeau 2002, 6–8) At the primary level, he argues, the digital object appears as inscriptions of signs on a medium, carrying no “morphology, syntax or semantics.” (Ibid., 6) This means that the interpretation of the bits is not defined at the level of inscription. (Ibid., 7)

At the second level, the “digital information object is a logical object according to the logic of some application software”, and thus at the logical level it is determined “how the input stream is transformed into the system’s memory and output for presentation.” (Ibid., 7) The logical aspect is in this regard tied to the computer’s ability to process the inscriptions according to some set of rules. This allows the inscriptions to be processed and transformed into something that is humanly meaningful — that is a conceptual object. Thibodeau describes the conceptual aspect as what we deal with in the ‘real’ world, “an entity we would recognize as a meaningful unit of information.” (Ibid., 8) The conceptual content can, however, be represented in a wide range of digital encodings, and hence, a similar output can be produced in many different ways. The content of a text document, for instance, “may be encoded digitally as a page image or in a character-oriented word processing document.” (Ibid., 8)

In terms of preserving digital objects, Thibodeau (Ibid., 7) states that although we have to preserve digital objects as physical inscriptions, it is insufficient in regards to preserving its function. In order to maintain its function, we need to consider the logical aspect of the digital object and know “the correct requirements for correct processing of each object’s data type and what software can perform correct processing.” (Ibid.,8) In addition, the conceptual object is not necessarily equal to the logical units that comprise it. For instance, a single text document in a word-processing format could consist of several subdocuments, leaving one single conceptual object stored as multiple logical objects. (Ibid., 11) It would thus appear as a single document to the reader, and as multiple documents to the computer. The conceptual object can in this regard be produced in many

different ways. The product is however always bound to the material components of the computer.

For the sake of clarity; there is a difference between ascribing conceptual properties to a digital object and asserting its immateriality. The ability to produce a conceptually similar object from a diverse set of states is not the same as it being liberated from its material or physical underpinnings. Even though the conceptual aspect can be encoded in a number of ways, it is always embodied in some kind of physical substance or phenomena. It is important to refute the notion of digital immateriality because such a belief may cause us to steer off target when we are trying to preserve these kinds of artifacts. To tackle the challenges of digital preservation, one must realize that the difficulty of ensuring these kinds of objects comes not from their alleged appearance in some kind of abstract non-physical space. Rather, the issues at hand are rooted in the dichotomy between machine and human language — a non-semantic and a semantic level — and in the complex workings of digital environments. To a human being, the microscopic bit inscriptions on the surface of an optical hard disk are not semantically meaningful. Issues related to digital preservation are, however, not merely caused by the encrypted nature of the digital artifacts themselves, but by the complexity of the encryption and the means required to interpret and translate the encrypted information. If I were to read a book written in Parsi, I would need a Parsi speaking human intermediary to translate it for me. However, in order to translate a bitstream, I would have to rely on a computer intermediary — and not just any computer intermediary — the right computer with the correct hardware setup, running a certain operating system that pertains to the file system needed to read the artifact. While the contents of a book may be directly available to someone, bits do not mean anything to anyone but a computer with the compatible hardware and software.

For a piece of software to function properly, we need to run it in an appropriate environment that complies with the dependencies of the executable file. This interdependent relationship between the artifact and its environment lies at the heart of the problem because the interdependencies between a piece of software and its environment cause another dependency — namely that between the human and the machine. By this, I mean that our access to digital artifacts depend on

having a specific computer intermediary that knows how to process the bits and output them for presentation. In turn, external factors such as obsolescence further threaten these artifacts as they remove our ability to access to the environment needed to process and output them.

## 1.2 Societal Challenges

### *1.2.1 Digital obsolescence*

Technological obsolescence is the process in which old technologies replaces new technologies, making the previous technologies outdated. (Harvey 2012, 51) There are different reasons why media becomes obsolete. For instance, the continuous improvement of technologies—such as the expanding of storage, increase in processing power, memory and so forth—makes the previous and less capable technologies outdated. This is a form of obsolescence due to innovation. Terry Kuny (1998, 2) claims that the cycle of obsolescence for information technologies averages at around 18 months. This estimate might be linked to a common misquotation of Moore’s law. Originally, Gordon E. Moore (1965, 2) estimated that the number of transistors on a computer chip would double every 12 months. Moore later attributed the estimate of 18 months to David House, the Intel executive at the time. (Kanellos 2003) House’s projection came close to the truth, as processors speeds were doubling every 20 months in the mid-2000s. (Ibid.)

The pace of technological progress is one of the factors that determine the rate of which technologies become obsolete. Every 20 months there will be twice as efficient chips on the market, making the previous generation of processors obsolete. This is a kind of obsolescence that is driven by scientific and technical innovation. Consider, for instance, the transition from 3,5 floppy disks to CD-ROMs, which increased the storage capacity from around 2 to 700 megabytes per unit. The increase in storage allowed game developers to implement high-resolution textures and make richer virtual worlds. The introduction of CD-ROMS had a rippling effect that changed many areas of the technological domain. Computer manufacturers replaced the previously standard 3,5 floppy drive with a

CD-drive, causing a decline in the market for floppy disks and drives. In addition, it also reduced the demand for cassette tapes. CD-ROMS could hold an entire album of lossless music and did not require the listener to change sides halfway through. The hassle of the tape curling up was no longer an issue, and the listening experience would no longer be distorted by the wows and flutters caused by unwanted speed variations in the rotary parts of the cassette player. It became possible to implement metadata such as song names and album covers, and it enabled the listener to easily switch between tracks. In other words, the CD-ROM technology was far more convenient than its predecessors, making the previous media less desirable, and in turn, obsolete.

Today, the CD-ROM appears to be on a path toward obsolescence. (National Archives of Australia 2016) The advances of the Internet have changed the way we acquire software — something of which is especially prevalent in the game industry. In the early 2000's it was still common to buy computer games on CDs and DVD-ROMs, but this changed with the introduction of Steam, PlayStation Store, Xbox Store and other web-based platforms. The era of buying boxed discs with pre-installed software is gradually coming to an end — a transition that can be attributed to the development of high capacity hard drives, in combination with faster broadband and fiber speeds. (Wingfield 2015)

Another form of technological obsolescence is the commercial model of planned obsolescence — a concept originally introduced by Bernard London in 1932. (Parikka 2015, sec. 2682) London proposed that the government should impose a tax on products that were being used beyond their intended lifespan, in order to spark activity in the economy and by that overcome economic depression. (London 1932, 2) Although the proposal was never implemented as a government initiative, product designers and the commercial industry adopted the idea as a way to boost their income by moving more products through the market. (Parikka 2015, sec. 2682) Thus, by intentionally shortening the lifespan of their products, and thereby shorten the cycle of replacement, businesses can increase their overall stream of revenue. This is also apparent at the micropolitical level of design. (Parikka 2015, secs. 2696-2697) By using plastic enclosures that are glued shut, proprietary cables and unchangeable batteries, manufacturers can restrict repairs and maintenance through the very design of their products. (Ibid. 2696-2697)

The principles of consumerism “propels producers and consumers into an increasingly virulent vortex of consuming and replacing.” (Serexhe 2013, 79) What is replaced thus become obsolete. Short replacement cycles pose a greater threat to the longevity of digital content than the deterioration of the hardware onto which it is stored. (Kuny 1998, 2) When companies move onto new lines of products without ensuring backward compatibility, go bankrupt, or simply halt the production and service of their products, the digital object is stuck in a limbo — trapped in an obsolete format or on an unreadable medium. (Ibid., 5)

The more dependencies a digital artifact has, and the more complex it is, the more vulnerable it is to obsolescence. If a book goes out of print, we do not lose access to the copy we already possess. If a million copies of the book were printed, chances are that at least some of them will survive for a hundred years or more. This is not true for computer applications. A piece of software has a multitude of dependencies, and these dependencies are shared by every single copy of the software. If one of these dependencies are unattainable, every instance of the application will be equally affected. If a specific driver or the operating system needed to run it becomes obsolete, none of the copies would be functional. Obsolescence thus adds to the vulnerability of software.

Whether driven by the market or by innovation, obsolescence reduces the lifetime of digital technologies and artifacts. This may cause a number of problems for archives. Say we are trying to recover and preserve a piece of software from an 8-inch floppy disk produced in the 1970s. First of all, the hardware needed to read the medium may no longer be available. Even if we can acquire a functional 8-inch floppy drive, its obsolete connectors are likely to be incompatible with the standards of modern computers. Let’s say, however, that we overcome this problem by using some kind of special adapter, and successfully connect the drive to our computer. What we now face is the problem of interpreting the bits inscribed onto the disk’s surface. As the software was made for an operating system that is now obsolete, our computer will not be able to understand what the bits mean. It is programmed differently from the computers of the 1970s, and the operating systems use different file systems — i.e. they organize, name, describe and retrieve files in different ways. (Kirschenbaum, Ovenden, and Redwine 2010, 15) Consequently, we have lost the means to interpret the bits, and thus lost access to

the functionality and performance of the software. In this regard, obsolescence causes problems at every stage of the archiving process — from the point of migration to the point of regaining access to the software’s functionality. In addition, obsolescence shortens the lifespan of digital artifacts and adds to their vulnerability. This provide new challenges for archives, as they have to preserve increasingly complex materials within a shorter time frame.

### *1.2.2 Legal issues*

The legal challenges of preserving software are many, and they vary from country to country. Although the specific legal issues encountered by Norwegian and U.S memory institutions will be discussed in more detail in chapter 3 and 4, there are some general issues that need to be addressed — regarding areas where legislation actively curbs preservation and cases where it threatens the longevity of digital materials.

In countries such as Norway and the U.S.A, software is protected by intellectual property and copyright laws. (Åndsverkloven, § 1; United States Copyright Office 2012, 1; Collin and Perrin 2013, 52) Copyright “gives a copyright holder the right to control the copying of that to which the right extends.” (Lessig 1999, 124) It governs the economic rights to copy, adapt, display, perform and distribute intellectual property. (Besek 2003, 2) Archives depend, however, on having these rights in order to counter the challenges caused by obsolescence. Firstly, they need to copy the materials and migrate them to safer storage systems. Secondly, the right to reproduce and adapt a work might be needed in order to reformat and update the files so that they can be read by newer computer systems and operating systems. Thirdly, an archive depends on having the ability to perform, display and provide public access to their collections in order to have cultural impact. The commercial need to protect intellectual property may thus cause problems for archives that want to preserve it.

Vint Cerf (Cerf 2015, 10:20-10:40), one of the fathers of the Internet, expressed his concerns about the conflict between copyrights and preservation rights in a lecture held at Carnegie Mellon University in 2015. He argued that copyright creates a number of problems for archives. It can prevent them from keeping and sharing copies of the software, restrict their ability to provide public

access to it and inhibit them from acquiring its source code. Cerf, therefore, concluded that digital preservation must have a place at the copyright table if our digital heritage is to be preserved. Neil Beagrie further echoes Vint Cerf's concerns. In his study of national digital preservation initiatives in Australia, France, the Netherlands and the United Kingdom, he found that “the needs of memory institutions for legal exceptions to undertake archiving are often overlooked or not sufficiently understood.” (Beagrie 2003, 3) None of the countries in the survey had comprehensive legal provisions for archiving digital publications, and he found that there was a general trend to increase the duration and strength of copyrights. (Ibid., 3)

On the one hand, copyright restricts the access to, and use of, intellectual property. If archives are not exempted from some of these restrictions, it can severely limit their ability to preserve copyrighted materials. Also, the duration of copyrights tends to exceed the lifetime of most digital materials, which is 70-120 years in the U.S. and 50-70 years in Norway (United States Copyright Office 2011b, 1; Åndsverkloven, § 40) As stated previously, the cycle of obsolescence averages around 18 months. Even though backward compatibility is often ensured for a longer period than that, history suggests that we will not have access to contemporary software in 50 years. Software that was made during the 1970s and 1980s cannot, for instance, be natively run on the operating systems we have today — and many of these titles are probably still protected by copyright.

In his book *Code*, Lawrence Lessig (1999, 124) states that copyright “is protected to the extent that laws (and norms) support it, and it is threatened to the extent that technology makes it easy to copy.” Thus, copyright is in a constant war with technology, as the development of the latter makes copying easier. (Ibid., 124) As a result, copyright holders are inclined to come up with ways to restrict access and use of their works. End-user License Agreements is one way of limiting the use of copyrighted works. Say you go to a hardware store and buy a hammer. After you have purchased it, the hammer is yours and you are free to do whatever you want with it. You are free to lend it to you neighbor, sell it or give it away. This tends not to be the case when it comes to software. When you buy a piece of software, you may not technically own it. Instead, you commit to an End-User License Agreement, an EULA, upon installation. What is worth noticing here is the word



license. Rather than owning a copy of the software, you own a license to use it within the terms set by its copyright holders. (Andris 2002, 1)

The distinction between owning and having a license to a copy of the software is important. § 117 (a) of the U.S copyright law states, for instance, that an owner of a software copy has the right to make an additional copy or adaptation for the purpose of maintaining its functionality or for preservation. (United States Copyright Office 2011a) By licensing rather than selling the software, copyright holders can nullify the rights given to the customer by § 117. Furthermore, the recent trend of software subscriptions, such as the Adobe Creative Cloud, further reduces ownership and puts limitations on access and use. Adobe can, for instance, remotely cut a customer's access to the software if the monthly fee has not been paid. (Pogue 2013) In this regard, the trend of licensing and subscription shifts the power balance from the user to the software companies, giving them increased control over their product. (Ibid.)

Another method for restricting use and access, are Digital Rights Management systems (DRM). This is a generic term for “access control technologies that can be used by hardware manufacturers, publishers, *copyright* holders and individuals to impose limitations on the usage of digital content and devices”, and is “used to describe any technology that inhibits uses of digital content not desired or intended by the content provider.” (Azad, Ahmed, and Alam 2010, 24) In relation to software, access control technologies include the use of license keys and dongles, online activation and limitations on re-installs. (Nützel and Beyer 2006, 368, 377; Azad, Ahmed, and Alam 2010, 26) Although DRM systems are targeted at preventing piracy, they also pose problems for archives. When recovering obsolete pieces of software, it may be difficult to acquire the appropriate license keys, and if the software is protected by online activation, the servers are likely to be inactive. DRMs may, therefore, remove archives' ability to migrate and copy the software and by that remove their ability to preserve it — especially if they do not have legal grounds for circumventing these access control technologies.

Making copies of a piece of software may not just be a question of whether or not you have the right to copy — it may also be a question of whether it is technically possible to make a copy. Even if archives are exempted from certain prohibitions related to intellectual property and copyright laws, they may still not

be able to make an archival copy of the software. The strengthening of copyrights, along with the software companies' fight against piracy and illegal use, can pose a serious threat to the longevity of digital materials and constrain archives' ability to preserve their collections. The continuance of our digital heritage thus depends on political decisions that ensure both the *rights* and the *means* to preserve digital materials.

### 1.2.3 Recognition

While obsolescence and strong copyright laws influence memory institutions' ability to preserve software, the failure to recognize the cultural value of software may limit both their ability and their will to preserve these artifacts. Due to limited resources and other restricting factors, archival methods involve a process of appraisal or curation. Within the traditional archival context, "appraisal is the process of determining whether records and other materials have permanent (archival) value." (Pearce-Moses 2005, 22) Appraisal decisions are made on the basis of a number of factors, including "the record's provenance and content, their authenticity and reliability, their order and completeness, their condition and costs to preserve them, and their intrinsic value." (Ibid., 22) Deciding on which materials to preserve are a necessary part of the traditional archival paradigm, "because there are usually more things — more information, more records, more publications, more data — than we have the means to keep." (National Library of Australia 2003, 70) Archives thus become authorities of memory, in the sense that their decisions influence the long-term cultural memory of a nation. They are the final gatekeepers of our cultural heritage.

When institutions decide on what to preserve, they are making value judgments. However, these judgments may be influenced by other factors than quality. Art historian Linda Nochlin (1988, 176) states, for instance, that the apparent lack of women among the great artists of the past, can be credited to the biases embedded in society and its institutions. It was, she states, "institutionally made impossible for women to achieve artistic excellence, or success, on the same footing as men, no matter what the potency of their so-called talent, or genius."

(Ibid., 176) Biases embedded in society may influence what reaches the archives, and it may influence decisions on what to include and exclude from the archives.

Institutional biases against new technologies, creative practices and genres may also be an issue. Before being admitted to archives, works of art and literature have usually received recognition within their respective fields and affiliated power institutions, such as galleries, publishing houses and critics. Acts of critique, curation and appraisal are attempts at deciding both the contemporary and the long-term importance and value of an artwork or cultural artifact. In this regard, it is a form of value prediction. This creates room for error, as the artifact may be of certain importance to the future that is not easily recognized within its contemporary context. In this case, the institutional bias is not related to gender or ethnicity, but rather rooted in the curators' standards of quality.

There is historical evidence pointing to a certain latency between the occurrence of new art forms and techniques, and the institutional acceptance of their value. New artistic practices may, for instance, be rejected by the establishment as they transcend contemporary frameworks for interpretation, classification and quality judgments. This is apparent if we consider the rise of the French Impressionist movement in the mid-1800s, and their polemical relationship with the art establishment. Each year, the French Academy of Fine Arts hosted the Paris Salon, a showcase of works that the jury of conservative artists and Academy members considered to be of the highest academic quality. (Visual-Arts-Cork 2016) The jury's standards for quality judgment adhered with a strict hierarchy of genres and works that belonged to a lower ranked genre would be regarded less favorably. (Ibid.)

Among the rejected the painters for the Paris Salon in 1863 were Edouard Manet, Paul Cézanne and Gustave Courbet (Ibid.) After increasing protests from artists, and the rise of counter-exhibitions intended to expose the bias of the academic juries, the Salon administration consented to the establishment of a Salon des Refusés — an alternative showcase featuring the artworks rejected by the jury. (Boime 1969, 412–13) The exhibit became a revelation for a young generation of artists, who saw its success as a confirmation of the validity of their own undertakings. (Ibid., 415) The success and publicity surrounding the exhibition, put pressure on the jury to sponsor it again the following year. However, the next year,

the leaders of the 1863 rejectees were deliberately admitted to the regular Paris Salon, so as to prevent a further repudiation of the Academy's authority. (Ibid., 415-416) Although the Salon des Refusés lost its sensationalism after the original exhibit in 1863, "the idea of the Salon des Refusés kept alive the hope of combating despotic juries and encouraged the aspirations of progressive artists" (Ibid., 416) According to Albert Boime, (Ibid., 417) Cézanne later implied that the jury of 1863 had failed to understand the young Impressionists, and noted that jury's verdict was irrelevant because it was based on an entrenched stylistic outlook. The Impressionists had offered an alternative style of art that the jury was simply unqualified to judge. (Ibid., 417)

Similarly, art museums "largely ignored digital and electronic art until the early 1990s." (Tribe and Jana 2009, 21) Although some established art institutions and foundations supported and showed an interest in computer-based art throughout the 1990s, it was mostly fronted by specialized institutions, online communities and small non-profits. (Ibid., 23) This was partly due to a "widespread skepticism among the proponents of more traditional forms, like painting and sculpture", and partly because of the anti-art-world attitude among New Media artists. (Ibid., 21-22, 24) Despite the increasing support during the late 1990s, there still remained "many arts funders and other established institutions that were not interested in New Media art". (Ibid., 23)

Jason Scott (2016, transcript, 7) notes in an interview with the author that there is also an institutional reluctance to acknowledge computer games as culturally important artifacts. Egenfeldt-Nielsen et. al. (2013, 157) ascribes this reluctance to new media skepticism. Similar to radio, film, and television, which have all fought against the perception that their role was to entertain rather than enlighten, computer games have not yet been acknowledged and accepted as culturally significant forms of expression. (Ibid., 157) In other words, Western societies tend to categorize computer games as pop culture, and thus deemed less worthy of attention than traditional high cultural forms of expression, such as literature and painting. (Ibid., 158)

What we can learn from this is that our institutions of both memory and recognition are not infallible. They make judgments as to whether or not a work has long-term value, and in their nature, these judgments are attempts at looking into

the future. As this is uncertain territory, acts of prediction often involve the risk of overlooking something, or simply make a wrong decision. As shown by the example of the Salon de Refusés, assessments of value are influenced by non-fixed standards of quality. Thus, institutions may not be able to recognize the value of new forms of expression within a contemporary context.

The question of whether or not software-based genres of expression can be defined as fine art or high culture is not necessarily important in relation to whether or not they should be preserved. Archives, libraries and museums come in many different forms and target a wide variety of media and information types, independent of whether they are perceived as low or high cultural artifacts — aviation museums preserve planes, computer museums preserve computers and so forth. Even though archival institutions differ in their specializations, they are usually aimed at preserving some aspect of the cultural memory. Their function is to purvey information about some previous or contemporary context. However, for software-based genres to be preserved, they must first be acknowledged by the institutions of memory and recognition. They must be recognized as cultural artifacts that are worthy of preservation. The point is, however, that these institutions are not infallible. Memory institutions are institutions of exclusion. When we choose to preserve something, it is always at the expense of something else. (National Library of Australia 2003, 70) Thus we need to be careful of what we exclude and make sure that our institutions are open for discourse. Although there needs to be some kind of selection process that precedes preservation, appraisal should be done at the level of individual works *within* a genre — not on the level of entire genres, artistic practices or the media used. It is thus important that archives are self-critical and try to avoid biases related to new media skepticism. If digital art, literature and games are to be preserved, archives must look past both the mainstream and the institutional skepticism toward these genres.

The failure to acknowledge the output of software-based artistic genres as important cultural artifacts puts these materials at risk. The limited lifetime of digital artifacts makes them far more vulnerable to time than previous forms of expression. While an overlooked novel can survive for centuries if kept away from harm, digital artifacts cannot survive through benign neglect — the storage medium may survive, but the ability to read it will be lost due to obsolescence. If entire

fields of digital artistic practices are overlooked by society and its institutions, chances are that the works will be lost. New media skepticism thus contributes to the problem, as by the time we have come around to recognize the value of a digital artifact, we may have lost access to it.

## 2. The whys, whats and hows of software preservation

### 2.1 Why preserve software

Jan Assmann (1995, 129) uses the term cultural memory to describe the collective remembrance of fateful events of the past that are maintained through cultural formation and institutional communication. In other words, texts, rites, monuments, recitations and practices — i.e. figures of memory— are what constitutes the cultural memory. He ascribes the term cultural memory with several characteristics. (Ibid., 130) Firstly, it involves what he calls the concretion of identity, namely the sharp distinction between that which appertains to oneself and that which is foreign, between those who belong and those who do not.

Secondly, the concretion of identity is connected to a capacity to reconstruct. (Ibid., 130) Although the cultural memory is fixed in figures of memory and stores of knowledge, every contemporary context relates to them differently. Thus it exists in two modes — in “the mode of potentiality of the archives whose accumulated texts, images and rules of conduct act as a total horizon”, and a “mode of actuality, whereby each contemporary context puts the objectivized meaning into its own perspective, giving it its own relevance.” (Ibid., 130) Further, it is characterized by formation, meaning that the collective knowledge is given form and made tangible through a wide range of media — including writing, pictorials and rituals. (Ibid., 131) This objectivation of collective

knowledge “is prerequisite to its transmission in the culturally institutionalized heritage of a society”. (Ibid., 130)

Cultural memory, as defined by Assmann, is “formed by symbolic heritage embodied in texts, rites, monuments, celebrations, objects, sacred scriptures and other media that serve as mnemonic triggers to initiate meanings of what has happened.” (Meckien 2013) Drawing on this concept, the preservation of cultural artifacts is important because they constitute and trigger our cultural memory. Thus, there is a strong link between objects, heritage and legacy. UNESCO, (National Library of Australia 2003, 28) defines heritage as our legacy from the past — as something that is, or should be, passed on to future generations on the basis that it is valuable. Furthermore, cultural heritage is often embodied in, and triggered by, objects. Sure there are forms of intangible cultural heritage that are passed down through verbal or corporal communication, but this is not what we fill our museums, archives and libraries with. The long-term cultural heritage is kept alive by objects that are being preserved, experienced and discussed, whether it be archeological artifacts, books, films or audio recordings. What makes a cultural artifact important, is that it embodies information about the society in which it was produced. It provides us with clues as to who we ones were and to whom we have become; it gives us insight into the human condition, our history and ways of life. It embodies knowledge about the evolution of politics, ethics, science and technology — it tells stories of both progress and regression.

With the uprise of the Information Age, more and more of the artifacts we produce are in digital form. This means that parts of our cultural heritage are now embodied in digital objects, for which computers are the main tool for both creation and presentation. (Ibid., 13) Many writers have transitioned from pen and paper to word-processing software, and video editors have mostly moved away from using guillotine cutters and tape in the editing process. In addition to remediating previous technologies and genres, digital technologies have also laid grounds for new forms of output, such as computer games, software art and electronic literature. The transition from an analog to a digital culture has caused a shift in the materials we produce. More and more of the things we create are digitally born, and never leave the digital space. The continuation of our digital cultural heritage thus relies on having rigid frameworks for preservation in place.

Bernhard Serexhe states that for several millennia, “the self-image of our society has been characterized by the emphasis placed on the durability of stable systems of cultural transmission.” (Serexhe 2013, 77–78) Digital technologies have, however, made it harder to maintain durable and stable systems of cultural transmission. With all their layers and dependencies, it is safe to say that digital artifacts pose a previously unmatched challenge for archival institutions — especially when considering that the technical and societal factors involved makes them far more unstable and vulnerable to time. As Serexhe states, human expressions are increasingly embodied in binary code — “a code that can no longer be read directly by the human senses, but only by machines which then render it in a form that is comprehensible to people.” (Ibid., 78)

In light of Serexhe and Assmann, one can say that the transition from analog to digital culture entails a future in which our self-image and cultural memory will in part depend on the successes and failures of today’s efforts to preserve digital artifacts. Although there has been established fairly rigid frameworks for preserving simple digital objects such as text, image, audio and video files, there are a number of challenges to be overcome in relation to the long-term preservation of more complex and dynamic digital artifacts, such as software. Establishing long-term frameworks for preserving them is therefore prerequisite to the continuation of our cultural memory and our digital heritage.

In the *UNESCO Guidelines for the Preservation of Digital Heritage*, digital heritage is defined as being “made up of computer-based materials of enduring value that should be kept for future generations.” (National Library of Australia 2003, 27) These materials include digitally generated artworks, commercial and non-commercial computer games and software applications. (Ibid., 29-30) Software-based genres of digital art and literature, and computer games, are in this regard part of our digital cultural heritage. In addition, I will argue, software can be valuable historical documents and be a prerequisite for the preservation of other digital objects.



### *2.1.1 Software as artistic media of expression*

Technological development brings new tools for artists and authors that both reform their methods of creation and the form of their output. Just as the development of optics and film laid grounds for new art forms of photography and cinema, the rise of digital technologies has led to new genres of art, literature and entertainment. Software are not mere tools for artistic practices. They can themselves be works of art, literature or entertainment. First of all, it is possible to argue that these have intrinsic worth, in the sense that they embody the expressions of the artists, authors and developers that create them. In order for archives to preserve these works, they must however also have extrinsic worth. By that, I mean that an artifact must be important to others before it is appraised by an archive. In other words, they must have cultural value and impact.

Software-based art, gaming environments and virtual worlds are cognate disciplines in their own right. (Delve and Anderson 2014, xxxix) Executable art has “dedicated artists, museums, techniques and commissioning procedures”, and “gaming environments and virtual worlds have their own games developers, games museums, conferences for the gaming community, etc.” (Ibid., xxxix) The artistic practices that produce software include digital art, literature and computer games. These genres sometimes overlap and are somewhat difficult to define. Digital art and literature may have ludic elements, and computer games may provide visuals and narratives of intellectual and artistic value. The intersectioning can often be credited to the multimodal nature of these works, meaning that they implement many forms of communication resources, such as text, images and sound. Furthermore, using software as a medium allows the artist, author and game maker to create interactive worlds that respond to a user's input. It permits the creation of dynamic, responsive, interactive and even randomized experiences.

Digital, or electronic, literature does not simply mean digital text. Scanning a printed book does not make it digital literature, nor does the act of writing a novel in Microsoft Word and publishing it as an e-book. Digital literature cannot be remediated or extracted from its digital environment without losing some of its properties and literary qualities. According to Scott Rettberg (2014, 167), what is meant by electronic literature is that “the computer (or the network context) is in

some way *essential to* the performance or carrying out of the literary activity in question.” Texts can, for instance, be generated by a computer, either through interaction or by some authored algorithm, making the work intrinsically dependent on the digital environment. Furthermore, the field “tests the boundaries of the literary and challenges us to re-think our assumptions of what literature can do and be.” (Hayles 2007, sec. 1) Subgenres of this field include generative stories and poetry, codework, kinetic poetry, hypertext fiction, interactive fiction, computer art installations and more. (Rettberg 2014, 172; Hayles 2007, sec. 2)

Digital art, or New Media art, share many of the qualities of digital literature, in the sense that the artworks cannot be separated from its digital environment. In other words, the artistic properties of these works depend on computation in order to be realized. The global New Media art scene emerged during the 1990’s and drew artists from other disciplines such as painting, conceptual art, performance art, activist art and cinema. (Tribe and Jana 2009, 10–11) Due to the advances in information technology and the birth of the Internet, artists began experimenting with computer-based art. (Ibid., 11) This led to the emergence of sub-genres such as Net art, software art, hacktivism, ASCII art, glitch art and others.

Digital art and literature do not match the widespread impact of computer games. Rather, these cultural forms are important on the basis of their artistic and intellectual value. In addition, they are important to the history of literature and art. Computer-based art does not exist separate or parallel to the art movements of the past, nor did it “arise in an art-historical vacuum.” (Ibid., 7) The communication techniques of digital art and literature can be traced back to 20th-century art movements such as Futurism, Dadaism, Surrealism, Pop Art and Video Art. (Ibid., 7-9; Manovich 2003, 22; Simanowski 2010, 18) It is an extension of this art historical branch, and thus part of art history.

The genre of computer games is a cultural force that has both shaped and defined digital culture. In 2015, 42% of Americans played video games for 3 or more hours per week. (ESA 2015, 2) The data showed that the average game player was 35 years old, and out of those who played regularly, 44% were female and 56% were males. (Ibid., 3) Similarly, a study from 2012 found that 53% of the Norwegian population aged 16-64 years old had played a computer game during

that year. (ISFE 2012, 4) Out of the 53% who played, 46% were females and 54% were males. (Ibid., 4) Another study carried out by the Norwegian Media Authority, found that 94% of all Norwegian children aged 9-16 years old played computer games. (Medietilsynet 2014, 9) In the survey, 98% of the boys and 90% of the girls stated that they played computer games in their spare time. What these numbers tell us is that computer games do not constitute a subcultural phenomenon. Players are found in all age groups and they are fairly balanced in regards to gender. Furthermore, the Norwegian numbers show that computer games are widely played by the upcoming generation. It is therefore reasonable to assume that this cultural form will have an even greater impact on human culture in the future.

If software is not preserved, we risk losing access to the creative works that exist in this form. If access to these works is lost, the legacy of the software-based art and literature movements of the 21<sup>st</sup> century will also be lost. Consequently, this would leave gaps in the history of both art and literature. Similarly, if computer games are not preserved, we would lose access to the works that helped shape and define the digital culture of the 21<sup>st</sup> century.

### *2.1.2 Software as historical resources*

Neil Chue Hong (2014, 112) states that software can be valuable historical resources and thus have inherent heritage value. A piece of software could, for instance, be the first of its kind or be a fundamental part of a historically significant event — and as historical resources they should be preserved for the future.

(Ibid.,112) Another argument for why we should preserve software is that they are resources for learning and for understanding the technological development. Many academic fields may have an interest in preserved software as they are necessary for understanding the technological development. By having access to obsolete software, students can learn about previous programming languages, trends in design, interfaces, mechanics and more. Thus, by exploring these programs, they can get a deeper understanding of the history of computing. Richard Allen Bartle argues that we should preserve computer games “because it allows the designers and critics of the future to understand how things got to be how they are, so they can better understand the trajectory of where they are going.” (2014, 17) Only by

knowing the rules, he states, “can you willfully break them; only by understanding what has been said can you disagree; only by reading can you learn to write.” (Ibid., 17) What he is pointing out, is that technological development is a discourse that is built upon previous achievements. By understanding, discussing and critiquing what was before, we can arrive at something new.

Further, Vint Cerf (2015, 13:40) argues that software preservation is important to ensure scientific reproducibility, and supports his claim by referring to a real world example. In 2010, a highly influential empirical study concluded that “economic growth declines dramatically when a country’s level of public debt exceeds 90% of gross domestic product.” (Ibid., 13:40) This study was cited by several EU governments and used as a basis for austerity measures. In 2013 a UMass graduate student found a bug in the spreadsheet software that had been used to perform the calculations. When he performed recalculations using the researchers’ dataset, he found that the growth was 2.2 per cent rather than -0.1 per cent. (Ibid., 13:59) If access to the software had been lost, it is likely that this error would not have been discovered. In this regard, another argument for software preservation is that it allows us to step back and re-analyze both the software itself and the data it produced.

In addition to ensuring scientific reproducibility and providing the means for understanding the technological development, software can also be used to document history. An example of this is the *New Dimensions in Testimony* project, undertaken by USC Institute for Creative Technologies. The aim of this project is to create a piece of software that implements 3D video technology, voice recognition and natural language processing in order to capture and tell the stories of Holocaust survivors. (Maio, Traum, and Debevec 2012, 23–24) The recorded stories and answers are added to a database, and AI technology is used so that a person can “engage in an interactive dialogue with the survivor, asking questions in their own words and hearing the survivor’s response.” (Ibid., 24) In other words, this piece of software purveys and captures historically important information, and thus is an important historical document. The experience produced by the software allow for a personal and direct access to history, long after the last survivors have passed. That is, if the software is preserved.

### *2.1.3 Software as a prerequisite to the preservation of other digital artifacts*

Digital preservation efforts are often targeted at what is considered to be the end-products of information, such as images, texts audio and video. (Hong 2014, 111) However, the software needed to access this data is frequently overlooked in the preservation process. (Ibid., 111) If we fail to preserve the software needed to translate the digital objects to a human-observable form, we risk undermining the very reason for archiving the materials in the first place. (Ibid., 111) Preserving software is thus “a prerequisite to the preservation of other electronic objects.” (Matthews, Shaon, and Conway 2014, 125)

Digital data can not always be preserved separately from the software needed to process it. This is sometimes true for scientific datasets that depend on spreadsheet software for calculations. It may be possible to extract the raw data, but the performed algorithms and calculations may be lost. Digital objects that contain scientific data can therefore not always be separated from the software to which it pertains. In some cases, this also holds true for digital artifacts created by artists. Some archives are increasingly receiving born-digital content from artists, which depend on proprietary software for both creation and access. (Allen 2013, 24) Maintaining access to these works, thus depend on having access to copies of the software that can read the work’s file format.

Say we want to archive the compositions, techniques and workflow of a contemporary electronic musician that produces music exclusively through the use of computer software. This could serve as an historical documentation of the techniques involved in the music practices of the early 21st century, and provide greater insight into the musicianship of this specific artist. In order to acquire this knowledge, it would not be enough to render the consolidated output of a song in WAV or MP3 format, as this would tell us little about how it was created. Now, let’s say that the musical composition that we are preserving was made with Ableton Live — one of the most widely used software for music production. What we would want to archive is the project folder, which contains the project file, instruments and sound files used in the composition. The first problem we run into is the format of the project file. Ableton Live’s project files are stored with the *.als*

extension, which is a proprietary format locked to the Ableton platform. This also holds true for the instruments, effects and presets native to the platform, which are stored with the *.adg*, *.agr* or *.adv* extensions. (Ableton 2016) As all these proprietary formats depend on the Ableton software to interpret it, we need to preserve the software in order to maintain access to the project file and the work it contains.

All digital files depend on a software intermediary to translate it. However, some information is more easily translated so that it can be read by other software than that which it was intended for. If we are preserving a text written in Microsoft Word, we can ensure future access to the information by extracting or converting the text to another format before Word and its formats become obsolete. This is not as easy when it comes to digital objects that contain complex information, such as heavy formatting, calculations, images, sound, video and external dependencies. Furthermore, companies may have an interest in using closed proprietary formats, so that only their own software product can be used to open the files. In these cases, digital artifacts cannot always be preserved separately from the software that reads it. Preserving software may therefore be necessary in order to maintain access to other digital objects.

#### *2.1.4 The imperative to preserve software*

When dealing with short-lived artifacts, such as software, time is the most critical factor to our ability to preserve them. Jason Scott (2016, transcript, 7) states that by the time we get around to recognizing the cultural value of computer games, we may no longer have access to them. The time between the recognition of an art form and its preservation must be shorter than the life-span of the artifacts they intend to preserve. Even though all materials deteriorate, digital artifacts tend to deteriorate at a faster rate than the artifacts traditionally targeted by archives.

Memory institutions have an obligation to provide the future with the information of today and to ensure the continuance of our cultural memory and our cultural heritage. In the Information Age, this means that they must purvey information about a digital culture and the artifacts it produced. Throughout history, eras have been defined by the tools and technologies that were available —

e.g. the Stone Age, Bronze Age, Iron Age, Industrial Age etc. Most of the human world today are defined by a widespread appropriation of digital technologies and tools. We use them as tools for production, science, communication, artistic expression and recreation. In order to capture and purvey information about the contemporary context, memory institutions must preserve these outputs. Their role is not just to preserve artifacts that are considered to be of high cultural value; they should also preserve artifacts on the basis that they carry important cultural information. Although software, such as computer games, may or may not be considered high-culture, they are undoubtedly a cultural force that has had a widespread impact on contemporary culture.

The imperative to preserve software can be contended in several ways. First of all, a digital culture expresses itself through digital media. This means that the legacy of the 21<sup>st</sup> century will partly be embodied in digital objects. If we agree to the premise that cultural heritage is something that should be preserved, we are left with an imperative to preserve the artifacts in which it is embodied. Today this includes software. Executables can be pieces of art, literature and entertainment, they can be important historical artifacts, they can hold valuable scientific data, and they are prerequisite to the preservation of other digital artifacts. Thus we should not dismiss software as culturally important artifacts. If left out of our archives and memory institutions, it could lead to irreparable holes in our cultural memory.

## 2.2 What aspect of an executable artifact should we preserve

When preserving software, it is important to have in mind that they are composite objects. Going back to the terminology of Thibodeau (2002, 6–8; National Library of Australia 2003, 35), a digital artifact is formed by an inscriptional, logical and conceptual level. At the level of inscription, the artifact is a series of signs (bits) inscribed onto some kind of surface. The logical level is where the bits are interpreted, processed and outputted for presentation. Finally, the conceptual level of a digital artifact is the level of presentation, where the artifact is recognized as humanly meaningful information.

An analog artifact such as a book is written and read using natural language. This means that there is a semantic correlation between its inscriptional and conceptual content. If we preserve the inscriptional level we also preserve its conceptual content. Digital artifacts, on the other hand, consists of non-semantic inscriptions from which the conceptual aspect is provoked through the computer's logical processing of the bits. This also means that there is a different kind of authorship involved in the creation of software than with analog forms of textual media. While a novel has a direct authorship, in the sense that the author writes directly to the reader using natural language, a piece of software is characterized by a kind of indirect authorship in the sense that the programmer writes to a computer intermediary in the form of code. By using a formal language, i.e. a programming language, the text can be processed by the computer and outputted for presentation. In this regard, there are two different levels of authorship involved in a specific piece of software — a back-end (written code) and a front-end (function/experience) The creative work of a developer spans both these levels, as she writes code so that the computer can produce the conceptual content she has envisioned.

Thus there are two levels of authorship, namely the authored code and the function of that code — i.e. the actual code written by the developer and the conceptual content that the code is made to produce. However, when a game developer creates a game, the event produced by the code may be more important artistically than the code itself. For digital artists and writers on the other hand, the authored code may serve an artistic point. The overarching question here is perhaps, what aspect constitutes the *work* in a piece of software? Is it the dynamic event produced by the processing of the code, is it the code in itself, or both? From an archival point of view it is important to reflect upon these questions, as the answers to them may influence which preservation strategies are viable and which are not.



### 2.2.1 Two levels of authorship — authored code versus intended

#### *function*

As should be clear by now, the conceptual content of an executable artifact is the produced output that we deal with on the screen or through some other kind of presentational device. For a game like *Super Mario Bros.*, the conceptual content is the responsive environment we interact with. It is the dynamic event produced and controlled by some set rules in the underlying code. Although the actual experience of the game may vary between players, they will all recognize the game as being an instance of *Super Mario Bros* through its narrative, design, game mechanics and sound. In other words, it is recognizable through its conceptual content. If all these elements are preserved, it means that we have successfully preserved its conceptual aspect. If the code is perfectly translated, a player would not be able to distinguish between the experience produced by the original code and the reformatted code if subjected to a blind test. In regards to preservation, it could thus be argued that the experience of the software can be preserved independently of the unique code written by its developers.

Such an argument could further be supported by pointing to a significant difference between authoring code and writing a novel, namely that the writing style of a programmer resembles that of a logician or a mathematician more than that of a novelist. By this I mean that a programmer writes logical strings that define certain rules for how a computer can produce an intended event. This poses the question of whether the work of a programmer is constituted by a specific document of code or the rules that it embeds. If we consider *Figure 1* on the next page, we see two pieces of executable code that both produce the sentence 'Hello World!' when run. Say that a programmer created this simple program and that he originally wrote it using the Java programming language. Now let us say that the Java programming language is at risk of becoming obsolete. Would it be a viable method for archives to translate the code to Python in order to preserve it? This would not have any impact on the end of the user, as the rules set by the developer would be logically preserved and thus produce the same conceptual content. However, it would transform the actual text written by the programmer. Thus we are left with the question of whether archives should treat executable code as a set

of transmutable logical strings or as a unique document of authored code — that is, should archives consider the *work* in the *Hello World!* program as a function that can be expressed through different logical statements or should they regard the work as inherent to the Java code written by the programmer?

Hello World!	
Java	Python
<pre>public class Main {     public static void Main()     {         System.Console.WriteLine("Hello World!");     } }</pre>	<pre>class Main {     public static void main(String[] args) {         System.out.println("Hello World!");     } }</pre>

Figure 1: Hello world program represented in the programming languages of Java and Python. (Learneroo 2016)

The answer to this question may depend on the genre or type of the executable, and it may vary according to the archive's reason for preserving it. If for instance, we are preserving a piece of software that is prerequisite to the preservation of other digital artifacts, such as a word-processing program, the main point would perhaps be to preserve its functionality. If we are preserving *Microsoft Word* for the purpose of maintaining access to the files that depend on it, it may not be especially problematic to alter the code so that it is compatible with newer environments.

For works of digital literature or New Media art, it may however be paramount to preserve the code in its original state. This is especially true for works like Nick Montfort's renowned series of digital poetry generators, named *ppg256*. These works were written using the Perl programming language, and are executable pieces of code that generates poems without the aid of any external dictionary, word list or other types of databases. (Montfort 2008) The intention behind the project was to make minimalistic poetry generators that could generate lexically and syntactically correct poems with the least amount of code possible. (Ibid.) This was done by distilling the English language into bigrams, so to form a database from which the software could generate words and sentences. (Ibid.) The series consists of seven poetry generators, where each work reduces the total

characters of code by one. In this regard, what we are preserving is a work in which the code serves an artistic point. If we take a look at *Image 1*, we see the output produced when the code of *ppg256-1* is executed. The generated poems may not make sense or be semantically sound, but hidden in its code is an artistic experimentation with the English language. By using code as a media for experimentation and artistic expression, it also becomes part of the work's conceptual content. The code is read and critiqued and is a prime target for discussion and our understanding of the work. In order to capture and preserve the literary and artistic properties of the work, it is therefore necessary to preserve its code.

```
the digs
moms of hale
caar on bunk
lang at baar
cows to toke
hast at bure
raed at habs
hons to boed
cont to mied
mods on boat
tale no baws
```

*Image 1: Generated output of Nick Montfort's ppg256-1*

### *2.2.2 Authorship and authenticity, and their impact on software preservation*

As previously mentioned, the conceptual content of a digital artifact can be reproduced and represented in many different digital encodings. (Thibodeau 2002, 8) In other words, a digital artifact can be changed on the logical level without altering how that artifact appears to the user. Reformatting simple digital objects, such as text documents and images, is therefore a common method used by digital archives to maintain access to the conceptual content over time. (Harvey 2012, 90; Cullen et al. 2000, 45; Riksarkivet 2010, 7) Also the *UNESCO Guidelines for the Preservation of Digital Heritage* states that it may be necessary for archives "to change the logical encoding so that it can present the same conceptual object using different technology." (National Library of Australia 2003, 36) This involves transcribing information to standardized preservation formats, so to avoid the trapping of information in obsolete formats. A JPEG image can, for instance, be reformatted to a TIFF image, without altering its conceptual content.

In terms of preserving software, the digital archival method of conversion and translation is more problematic. Firstly, reformatting complex digital objects is

technically difficult. More importantly, conversion of code poses an important set of questions in relation to authorship and authenticity. While the archival method of conversion and reformatting is based upon the idea that the conceptual content of an image exist independently of its format, and thus can be reproduced in any suitable encoding, an executable artifact cannot as easily be separated from its underlying code. In contrast to a digital photograph, in which the craftsmanship mainly lies in its conceptual content, i.e. the composition, colors, contrast etc., the craftsmanship of a computer program consists of both written code and the conceptual content that it produces. The authenticity of a piece of software can therefore not be entirely separated from how it is encoded, as the code is part of the authorship.

Preservation methods that alter the code of a work are not viable methods for preserving software, as it directly affects the authorship of the work. This is especially true for executable works in which the code serves an artistic point, but it is also true for software in which the functionality of the code is deemed the most important. Although there may be cases where the function or experience of the software may be considered more important to preserve than the exact code written by its programmer, archives must be aware that the original code may be valuable to a wide range of people, including critics, engineers and researchers. As discussed in section 2.1, executable artifacts are also historical resources. Regardless of which aspect serves the artistic point, the code is a valuable piece of information. For a game researcher, the original code of *Super Mario Bros.* is a valuable historical document, as it allows for analysis of how it was originally programmed and the methods used. Similarly, the code of *Microsoft Word* may be valuable for researchers studying the historical development of word-processing technologies. In other words, it is necessary to preserve the originally authored code in order to allow for future research.

In an archival context, it is not sufficient to treat an executable artifact as a solely conceptual object. This is because the experience and functionality fail to account for historicity and the authenticity of authorship. Translating the authored code is to partly remove the programmer from the work. In addition, translation can also lead to slight changes in the produced conceptual content over time. Similar to translations of books, there is a chance of information loss if we translate the text back and forth over a long period of time. This makes it impossible for archives to

ensure the trustworthiness of the software and it makes it difficult to guarantee that it is what it claims to be. In this regard, access to the original data must be preserved in order to ensure the authenticity of the executable work. However, if the functionality and performance are lost we can not say that we have truly preserved a piece of software. A viable strategy for software preservation must therefore preserve functionality without changing the authored code.

## 2.3 How to preserve software

In his seminal article “Ensuring the Longevity of Digital Information”, Jeff Rothenberg (1999, 11) uses the terms transcription and translation to describe two different approaches to digital preservation. Throughout history, information has been refreshed by copying it in its original language or by translating into whatever language is current. (Ibid., 11) Ancient texts recorded on parchment rolls have, for instance, been copied to other media to protect it from the deteriorating fabric. However, as languages change over time, we may lose the ability to understand what was once written on the parchment. (Ibid., 11) Even though we still have access to the text, it no longer purveys humanly meaningful information. In order to avoid this problem, it is useful to translate the information to a current language, so that it can be understood within a contemporary context. (Ibid., 11) The problem with translation, however, is that it involves changing the original text. While transcription retains the authenticity of the text and involves little to no loss in information, translation makes it difficult to determine whether information has been lost and makes it hard to track the changes that have been made. (Ibid, 11-12)

From a digital archival perspective, authenticity does not connote the Benjaminian concept of a unique first. (Benjamin 1936, 24; Kirschenbaum, Ovenden, and Redwine 2010, 10) All digital artifacts are intrinsically reproducible, and every time a program or a file is opened, the computer revokes a new copy of the artifact. In this regard, the digital object is always performed and reproduced by the computer. For digital archives, authenticity thus signifies the trustworthiness and integrity of the digital object — meaning that the information has not been altered and that it is what it claims to be. (Kirschenbaum, Ovenden, and Redwine

2010, 34) A viable strategy for software preservation does, however, depend on both protecting authenticity and sustaining functionality. Different strategies preserve different aspects of the software, and which strategy is most viable, depend on what aspects of the software we are trying to preserve. (Cullen et al. 2000, vi)

### 2.3.1 *Hardware preservation*

Hardware preservation is a holistic approach that preserves the digital artifacts by preserving their intended environment. In contrast to migration and emulation, it is an independent and standalone strategy for keeping software alive. By sustaining the life of obsolete computers, consoles and their operating systems, we also prolong the life of executables compatible with the platform. (Harvey 2012, 132) The benefit of this strategy is that it allows for a complete preservation of authenticity, both of the digital materials, and the hardware and operating system that renders them. (Ibid., 132) This means that all aspects of a digital artifact are preserved, from the level of inscription to the level of presentation.

Although hardware preservation is the best approach in regard to ensuring authenticity, it is not a viable strategy for long-term preservation. (Ibid., 132) Even though it extends the time in which we can access obsolete digital materials, obsolete hardware cannot be kept in a functional state for an indefinite amount of time. (Ibid., 132) In order to ensure the longevity of executables, it is therefore not sufficient to preserve the hardware platforms on which they run. As a general rule, software must be separated from its original environment to protect it from the obsolescence of hardware and operating systems. However, there may be some executables that cannot be separated from its intended platform without losing its integrity. This is apparent if we consider Cory Archangel's digital artwork *Super Mario Clouds* from 2002. Archangel made this piece by hacking the game *Super Mario Brothers*, which was originally made for the Nintendo Entertainment System in 1985. (Tribe and Jana 2009, 28) He replaced the programming chip in the original cartridge with his own chip, erasing everything in the game except for the pixelated clouds. (Ibid., 28) This work is constituted by both the software and the hacked hardware, as the media almost becomes a sculptural artifact, bearing obvious signs of being tampered with.

For software-based artworks that cannot be separated from its hardware without losing artistic value, hardware preservation may be the only proper method for extending their lifetime. Nevertheless, this is a futile strategy in the long run. (Rothenberg 1999, 15) When the hardware is no longer serviced and spare parts become increasingly hard to find, we will inevitably lose access to the hardware and thus lose access to the preserved software.

### *2.3.2 Migration*

Migration is a way of transcribing information, by “transferring digital information to new platforms before the earlier one becomes obsolete.” (Matthews, Shaon, and Conway 2014, 128) In other words, digital materials are copied from obsolete media and onto safer modern storage systems. The migration process thus eliminates the need to sustain and maintain the obsolete media onto which the materials are stored. (Rothenberg 1999, 15) In practice, this done by using a technique called ‘imaging’, which involves making a soft copy of the entire storage medium, rather than copying the individual files. (Kirschenbaum, Ovenden, and Redwine 2010, 71) One of the benefits of this method is that archives have to deal with a single image-file, rather than thousands or possibly millions of individual files. In addition, the produced image-file provides a protective container, as the operating system can interact with the image file rather than its contents. (Ibid., 31) In terms of authenticity, the migrated data is controlled by comparing checksums. These are mathematical values that are used to detect errors and to verify the integrity of the data. (Pearce-Moses 2005, 69) The checksum of the image file can thus be compared to the original bitstream, to ensure that an exact copy has been made. (Kirschenbaum, Ovenden, and Redwine 2010, 31)

Although migration prevents the bits from deteriorating along with the storage media, it does not protect the software from the obsolescence of the operating system and hardware needed to interpret the bits. (Harvey, 2012, 142; Huber, 2013, 142) Even though migration is a necessary stage in all long-term digital preservation strategies, it is not a sufficient method on its own. (Harvey 2012, 142) This is especially true for complex digital artifacts such as computer programs. Even though the authored code and the files are preserved, the ability to

execute the computer program will eventually be lost. A long-term strategy for preserving functional software thus depends on migrating the data *and* maintaining an environment in which it can run.

### 2.3.3 *Emulation and reconstruction*

The concept of reconstructing or remaking a piece of software accords with Rothenberg's concept of translation. The code of the executable work can be rewritten in a modern programming language so that it is compatible with newer operating systems. What is lost in this process, however, is the authored code. Even though it might replicate the output of the original programming, and by that replicate the conceptual level, the original code has been fundamentally changed. This does not necessarily pose a problem if the main point is to preserve the behavior or the produced output of the code. For someone who wants to play an early version of *Super Mario*, this may not pose a problem, as long as the game behaves, looks and responds in the same way as the original. However, for a person that wants to explore Nick Montfort's series of poetry generators, it is crucial to have access to its source, as the code itself serves an artistic point. Remaking or reconstructing executables, thus involve making changes to the actual artifact we are preserving.

Emulation works differently than reconstruction. Instead of rewriting the program, emulator software is used "to translate instructions from original software to execute on new platforms." (Thibodeau 2002, 19) While reconstruction involves translating and changing the original code, emulation uses a software intermediary to interpret the original code so that it is understood by contemporary platforms. In other words, this strategy relies on emulating the obsolete hardware and software needed to interpret the bits. (von Suchodoletz, Guttenbrunner, and Rechert 2012, 1) As a result, it comes close to reproducing the original look and feel of the work. (Ibid., 1) By using a software intermediary to translate the migrated data to a language that current platforms can understand and process, both the authenticity and the performance of the software is preserved. The authored code and the files are contained within a disk image, and the emulator can process this data and produce the output. Emulation is in this regard a subsequent stage of migration, as it is necessary to migrate the data in order to emulate its output.



An emulator that simulates a specific operating system can enable all software adhering to that platform to execute using the same emulator. (Thibodeau 2002, 19–20) Instead of having to reconstruct each piece of software, this strategy makes it possible to simulate an environment in which all compatible executables can run. However, emulators are also pieces of software and thus share the same preservational challenges as the materials that they are meant to preserve. (Huber 2013, 143) As emulators depend on certain operating systems to operate, they are themselves subjected to the dangers of obsolescence. (Ibid., 143; Thibodeau 2002, 20) The benefit is, however, that we would only need to preserve the emulators in order to preserve the function of all executables pertaining to the emulated platforms. As long as applications, such as computer games, are migrated and imaged, it will only be necessary to update and reprogram the emulators in order to keep these works functional.

### *2.3.4 Metadata, paratexts and documentation*

Metadata, paratexts and documentation provide context to the archived software. Providing context is a necessary supplement that accounts for the areas that the other preservation strategies do not cover, namely information *about* the software. This can be descriptions of how the artifact worked, how it was supposed to be used or other kinds of technical documentation and metadata. In addition, it may also be important to preserve the information that *surrounds* the work, accounting for the social and cultural impact of the work.

Digital objects are not normally self-descriptive. (Dappert et al. 2013, 107) Archives must therefore have access to information about the digital object, its environment and its dependencies to be able to use the artifact and to make it understandable in the future. (Ibid., 107) As a result, “metadata about digital objects’ computing environments must be preserved together with the digital objects as part of their core metadata.” (Ibid., 107) The widely adopted *OAIS* model for archiving digital materials, which is used by Riksarkivet, incorporates this kind of information. (Riksarkivet 2010, 12; Consultative Committee for Space Data Systems 2012, 2–6) Using this model, archives create an Information Package that contains both Content Information (the digital object/bits) and Preservation

Description Information. (Consultative Committee for Space Data Systems 2012, 2–6, 2–7) The latter contains the information archives need to be able to properly preserve the artifact, such as its processing history, external dependencies or relations, checksums, access rights and provenance information. (Ibid., 2-6, 2-7)

While metadata is necessary for an archive’s ability to preserve a work, other contextual materials can contribute to our understanding of it. When preserving a computer game, there may be a wide variety of contextual materials that should be archived in order to understand its social and cultural impact. (McDonough 2014, 189) In addition to the metadata containing information about its creator and platform, it may be beneficial to archive information about how it was received, how it was played, who played it and so forth. In order to accomplish this, we can preserve reviews, gameplay videos, screenshots, cover images, game wikis and other paratextual materials created by the developers and the game community. This is also important in order to allow for future research. Say that we fast-forward to 2066 and we are researchers conducting a historical study of early propagandistic computer games and their effectiveness. The games *America’s Army*, which was funded by the U.S army, and the Hezbollah produced game *Special Forces* could serve as examples in this case. (Machin and Suleiman 2006, 2; Zyda et al. 2003, 219) Even if we had access to functional emulations of these games, it would not tell us anything about its players, nor would it tell us much about how much they played and how they responded to the game. For instance, by just playing *America’s Army*, we would not know that the game had 1.3 million registered users out of which 800 000 completed basic training. (Zyda et al. 2003, 220) The games themselves do not provide information about their influence, nor do they capture the fan cultures, controversies and debates they trigger. Thus contextual materials are needed to map the areas that are not covered by the software itself.

Similar requirements apply to software-based works of New Media art and digital literature. (Konstantelos 2014, 241) In addition to the technical metadata needed to preserve the work, there may also be a need for detailed metadata and descriptions of how it worked. This would especially be true in cases where the executable is part of an art installation. In order to be able to exhibit these works in the future, it is necessary to know exactly how the software worked and how it controlled the installation objects. In addition, it is important to account for the

social and art-historical context of works of digital art and literature. In order to preserve the extrinsic value of the work, i.e. its social and cultural impact, we need to preserve both the work itself and its paratextual materials. This could include critical texts, close readings, art books, exhibition pamphlets and video documentation of people's interaction with the work.

### *2.3.5 Finding a viable strategy for software preservation*

Even though migration is not a viable strategy on its own, it is a necessary step toward emulation. Without migrating the data, we simply have nothing to emulate. The first step is therefore to image the media onto which the executable artifact is stored, and transfer the image-file to safe and modern storage systems. If we stop the process at this point, chances are that the operating systems and hardware needed to interpret the files will become obsolete over time. Migration must therefore be followed by emulation if we are to sustain the functionality and performance of the software.

None of the strategies manages to fully archive every aspect of a piece of software. An emulated game might behave differently than the original. The conceptual aspect of an executable artifact —the way it is perceived and experienced by the agent who engages with it— may be different from how it was originally experienced. If we play an emulation of a computer game from the 1980s, the devices for input and output may have changed. We may not, for instance, have access to the original controllers or the presentation devices it used to be displayed on, and there may be small changes in the functionality and the overall presentation of the work. Although the authenticity of the imaged software is kept intact, it may not be possible to fully preserve its conceptual aspect using emulation — i.e. the presentation and experience of the game.

In order to account for the aspects that emulation is not fully able to capture, we need to acquire metadata, paratexts and documentation. As much of these materials exist online, these can be gathered through web archiving. By using web crawlers, which are automated bots that traverse the Internet and fetches data, archives can gather contextual materials affiliated with a certain work. (Roche 1998, 102; Kulturdepartementet 2016a, 24) In conclusion, it appears that a viable strategy for the long-term preservation of software involves a combinatory

framework that implements migration, emulation and context. While migration makes sure that the bits of the software are safe, emulation preserves its logical and, to some extent, conceptual level. What is lost through emulation can partially be accounted for through metadata, paratexts and documentation, as they provide the basis for further understanding and knowledge of the work in the question.

## 3. Empirical analysis of U.S frameworks for software preservation

### 3.1 The Media Archaeology Lab



*Image 2: Desktop computers in the main room of the Media Archaeology Lab*

#### 3.1.1 What is the MAL

The Media Archaeology Lab was founded by Lori Emerson in 2009 and is located at the University of Colorado in Boulder. (Media Archaeology Lab 2015) The lab features an extensive collection of obsolete computers and digital media, and provide researchers, students and visitors access to what is now considered obsolete technologies and platforms. In addition to other hardware, such as

typewriters, word processors, fax machines and e-readers, the collection features 35 portable computers, 73 desktop computers, 22 handheld devices, 8 computing devices and 10 game consoles — spanning four decades, from the early 70's through the 00s. (Emerson 2016, e-mail message to author, October 12<sup>th</sup>)

During my conversation with Lori Emerson on April 12<sup>th</sup> 2016, she noted that the initial motivation for founding the lab was threefold. Being a researcher at the Department of English and the Intermedia Arts, Writing and Performance Program at the University of Colorado, her research can be situated within the field of digital humanities — spanning topics such as media poetics, history of computing and media theory. Emerson's interest in electronic literature and poetry, and her research on bpNichols early e-literary work *First Screening*, enticed her to found the lab, in order to preserve access to these early works of electronic literature. *First Screening* was a software-based digital poem programmed in Apple Basic that was distributed as a DSK file for the Apple IIe. (Flores 2012) The Media Archaeology Lab preserves this work in its original form, within its original environment, and allows visitors to experience the poem in the way it was intended by the author.

Secondly, Emerson founded the lab to deconstruct the consolidated and mainstream version of the history of computing. Rather than reducing history to a simplified timeline, Emerson (2016, Conversation) proposes that we see it as a branching tree. By doing so, one points out the dead-ends and could-have-beens of technological development. The point being that the success of specific computers, and a lack thereof for others, does not necessarily directly follow from the quality of the technologies themselves. Rather, there are multiple factors involved, such as timing, corporate power relations, marketing strategies and so forth.

Thirdly, the MAL is in itself an experiment — a research project that examines and dissects the operation of media archaeological collections. (Ibid.) It is a kind of meta-level project in which the lab itself becomes an object of research. In this regard, the MAL does not limit its efforts to the preservation of software as it also expands into other fields of research. It is a historical museum of hardware, and a place for retrospect, as well as an archive of obsolete works of digital art, literature and games. In addition to the extensive collection of digital computer systems, The Media Archaeology Lab also keeps documentation for each of these platforms and machines. (Emerson 2016, e-mail message to author, April 26<sup>th</sup>) This



from different fields of studies, and it is difficult to acquire a staff of volunteers that cover all the skill-sets needed to run and maintain the lab.

The dependency on volunteer work is a big challenge to overcome. Emerson pointed out that she is in the process of applying for large national-level grants to withstand the deficit. (Ibid.) However, the cost of running the MAL cannot only be measured by its monetary expenses — it is a time-consuming project that also eats into other parts of life. Emerson (Ibid.) notes that the time she spends on grant applications eats into the time she could have spent advancing her academic career by working on her book projects.

### **Maintenance and repair**

The problem of keeping the machines running is very much intertwined with the challenge of acquiring sufficient funds. Emerson writes (Ibid.) that labor is the most expensive and the most needed resource in order to ensure the longevity of the lab. At the present time, there is a lack of an onsite person qualified to perform repairs on the machines. Whenever a system breaks, they have to rely on finding someone who can fix it and who is willing to do it for free. (Ibid.)

The machines in the collection are old and fragile, and they are constantly breaking. (Ibid.) In addition, some of the machines acquired by the MAL are broken to begin with. Some of the systems they have purchased through vendors such as E-bay have arrived in a non-functional condition or in pieces. (Ibid.) The case of constantly breaking or dead-on-arrival hardware, combined with the lack of an onsite repair technician is therefore a serious threat to the MAL.



*Image 4: Storage room for broken hardware and spare parts*



Emerson (Ibid.) made it clear that the collection comes with a limited lifespan. Even with a technician working full time on the maintenance of the machines, there will come a time in which spare parts will be unavailable, and one would have to reproduce certain components to keep the machines functional. This could drive the costs up to a point where the sustenance of the lab would be an even more costly endeavor. Thus, it is probable that the MAL in time will become a museum of dead computers. Emerson (Ibid.) states that she does not expect the collection to be functional in 10-15 years. However, she is still optimistic;

Maybe the machines will surprise me and continue to work - after all, they're already working decades longer than they were meant to, or were designed to. (Ibid)

### *3.1.3 The importance and benefits of The Media Archaeology Lab*

One should not dismiss an institution like the MAL on the basis of its ephemerality. Even though it might be temporary, at least as a collection of functional obsolete machines, it certainly has value. It might not be a long-term solution for preserving software, but it excels at preserving its authenticity. The experience of reading the Javascript reconstruction of bpNichol's *First Screening* on a contemporary Apple MacBook Pro is not the same as reading it on the original Apple II. (Vispo 2006) The screen is flat, the dimensions are different, and one does not have to mount a 5.25" floppy disk to run it. In addition, the code itself is different from that which was written by the author. Even though the reconstruction successfully reproduces the semantic content, it is still different from the original work. This is where MAL excels, as it comes as close to offering the full authentic experience of the work as possible.

As previously stated by Emerson, the MAL is not just a place for the preservation of software. It is also a historical collection of hardware, showcasing the history of computing. This makes it a valuable resource for students, scholars and researchers from a wide range of academic disciplines. If the MAL, at some point in the future, does become a museum of dead machines, it would not lose its



status as a valuable learning resource. Rather, it would perhaps transition into being a museum of technological relics, telling a story of both obsolescence and the history of computing.

A museum of dead machines and storage media would perhaps play another substantial role. The importance of preserving even non-functional obsolete technologies became apparent with *The Lunar Orbiter Image Recovery Project* in 2007. The aim of this project was to recover images taken by NASA's five lunar orbiters between 1966 and 1967. (Bierend 2014) At the time, the images were stored as low fidelity prints, while the high fidelity tape originals were shoved into boxes and forgotten. However, in 2005, NASA engineer Keith Cowing and Dennis Wingo, a space entrepreneur, sought out to recover the high fidelity versions of the images taken by the lunar orbiters. (Ibid.) The tapes, however, could only be read by a special kind of hardware — the super-rare Ampex FR-900 drive. (Ibid.) Luckily, someone had kept a small number of them in a small shed in the outskirts of Los Angeles. Over the years the drives had, however, stopped working and “had to be rebuilt and in some cases completely re-engineered using instruction manuals or the advice of people who used to service them.” (Ibid.)

What would have happened if all the drives were gone, along with the original documentation? It is easy to imagine a gloomier situation, where both the hardware and the knowledge about that hardware was gone. Instead of re-engineering it, one would have to re-invent or make an entirely new and compatible system, in order to read the data from of the drives. This would be a challenging and costly project, depending on the complexity of the original system.

Institutions such as the MAL play a vital role in ensuring long-term access to the knowledge about obsolete computer systems and storage media. Preserving non-functional technologies is useful, as it enables us to re-engineer them and regain access to critical data. Although being a measure to be used only as a last resort, history has shown the necessity of having such a safety net. As the MAL preserves hardware, user manuals and other documentation, it makes the world less vulnerable to loss of scientific data and other culturally and historically significant information.

The value of having an institution like the MAL, should therefore not just be measured by its ability to keep its machines alive, nor by its capacity to provide

long-term access to live software. Although it is not a viable option for ensuring the longevity of digital content, it serves a spectrum of other important roles — it is both a mediator of our digital history and the last stance for our digital heritage.

## 3.2 The Born-Digital Program/Special Collections at Stanford University Libraries

### 3.2.1 *What is the Born-Digital Program*



*Image 5: The Forensics Lab at Stanford's Cecil H. Green Library and one of their capturing stations.*

The Born-Digital Program is part of Special Collections at Stanford University Libraries (SUL). The overarching aim of the program is to acquire and deliver historically significant born-digital material — ranging from academic articles to large collections of software and computer games. The material is captured and processed in the Forensics Lab, before being handed over to Stanford's Digital Repository for further preservation — the institution's own digital archive, providing access to a significant amount of research and archival material. (Stanford University Libraries 2016) For the most part, Special

Collections target and preserve research papers and text documents from faculty members and other renowned scholars. (Chan 2016, transcript, 8-9) However, even though they predominantly archive digital texts, they also have collections that contain a fair amount of software.

SUL owns a series of large and small collections of software, including the Stephen M. Cabrinety and the Steve Meretzky collection. (Ibid., 9) The Cabrinety collection is “one of the largest pristine historical collections of microcomputing software in the world, including titles from virtually all of the major microcomputer platforms...” (Olson 2013) From early on, Cabrinety had the idea of making a museum of software. As a teenager, he started collecting software and continued his efforts throughout his life. When he passed, the Cabrinety family donated the entire collection to Stanford - totaling at around 13 000-15 000 items. (Chan 2016, transcript, 8-9; Olson 2013)

SUL have a targeted practice of collecting software. Most of the material comes from collections that they have bought or that have been donated to them. In other words, there is a process of appraisal before accepting a collection. Instead of trying to gather everything, they buy or receive material that has already been deemed important. In addition, the material in their possession goes through a second curation process, in the sense that they focus their spending on the most historically significant items within these collections. (Chan, 2016 transcript, 2) Even though SUL handles a limited amount of software collections, the size of some them makes the preservation process a demanding and time-consuming task.

As of now, SUL is at a very early stage in the process of preserving their software. (Ibid., 5) Although much work has been done capturing and imaging the material, they have yet to move to the second phase, in which the static image-files are re-animated and made accessible through emulation. They have, however, in collaboration with UC Santa Cruz, started planning for this transition, discussing emulation strategies and submitting grant proposals. (Ibid., 5)

### *3.2.2 SUL's framework for software preservation*

The first step in preserving a piece of software is to acquire a copy of it. (Ibid., 2) As mentioned earlier, SUL either go out and buy a collection of boxed software or it is donated to them by the owner. Due to the extensive amount of software, most of the capturing work is outsourced and done off-premise. For the Cabrinety collection, SUL has partnered with the National Institute of Standards and Technology, which is a government organization under the U.S. Department of Commerce. (Ibid., 2; The National Institute of Standards and Technology 2015) NIST has an interest in collecting checksums for as many software packages as possible, which in turn can be added to a database and used for criminal investigation and anti-piracy purposes. (Chan 2016, transcript, 2) The collaboration is thus of mutual benefit to both parties, as NIST offers to capture all of SUL's material in exchange for the checksums of the packages. SUL, therefore, sends all the original boxes of software to them, and NIST captures them and returns the images. (Ibid., 2) The failure rate has, however, been unexpectedly high. Out of the 8000 storage media, only 50% have been successfully recovered. (Ibid., 2)

Although they have a rigid framework for preserving simple digital objects — text files, images, etc — SUL is still at an early stage of preserving software. They do not yet have a final framework for the long-term preservation of their software collections. To ensure that the data is not lost in the meantime, it is paramount to get the material off of the original obsolete storage media, and onto safer storage systems. (Ibid., 11) As Peter Chan (Ibid., 10) points out, if one does not copy the data now, but instead waits five years, the risk of losing the content is far greater. The initial priority and first stage of preserving the software is therefore to migrate the data from the aged and high-risk storage media. This is done by making a complete digital snapshot of the entire storage media on which the software is stored, along with its original file structure. As a safeguard, the media is imaged several times, and the checksums are compared to ensure the integrity of the image-files. (Ibid., 11) The valid image file is then copied and stored in the Stanford Digital Repository.

Even though SUL is not yet at a point where they actively preserve the function of the software, they have bought themselves more time by migrating the bits. Copying the bits, or the code, is however not sufficient in the long run. (Ibid.,

11) The next step is to preserve the software as both a logical and conceptual unit by ensuring that the bits can be transformed into the computer's memory and outputted for presentation. (Ibid., 11; Thibodeau 2002, 7) In other words, if the digital information is to be preserved as a logical object, one has "to know the requirements for correct processing of each object's data type and what software can perform correct processing." (Thibodeau 2002, 8) In order to keep functional copies of software over a long period of time, it is thus necessary to preserve the bits along with the technical metadata and an environment that knows what these bits mean, and that is able to execute and perform the code. As Chan (2016, transcript, 11) notes, there are basically two main strategies for digital preservation — migration and emulation. However, with regards to ensuring the logical aspect of complex digital objects such as software, he states that emulation is the only viable solution for long-term preservation. (Ibid., 19) If you cannot play the game, he remarks, you cannot experience the software. (Ibid., 19)

Peter Chan (Ibid., 19-20) sees the EaaS emulation environment as the most promising platform, and SUL is planning to implement this in the next phase. They have also considered the more mature VMware platform, but have come to the conclusion that it is a less favorable choice. First of all, in contrast to EaaS, VMware is not open-source. (Ibid., 20) It is proprietary. Secondly, it is not possible to feed a disk-image directly into the emulation environment. Instead, one has to use VMware's external conversion software to convert the appropriate physical machine into a virtual machine. (Ibid., 19-20) The platform needs to know more about the hardware in order to run the disk-image smoothly. A normal bit by bit disk-image will therefore not work in the emulated environment. (Ibid., 20) EaaS is, however, a more promising platform as it offers a simplified front-end, and does not require that the user performs any software-to-hardware conversion. Basically, it is possible to feed a pure disk-image directly into the EaaS platform, and it will take care of all conversion back-end. (Ibid., 20)

So for SUL, the first step in ensuring the longevity of their software collections is to preserve the bits by imaging the media on which the software is stored. The second step will be to preserve the performance of the software and to reestablish access to it, through emulation. However, when talking about preservation "we should not forget all the things about a game or about a software." (Ibid., 11) Peter Chan stresses the importance of preserving documentation and

metadata. He proposes the idea of creating an international web-based registry for software preservation. (Ibid, 3) The benefits of this are twofold — firstly, it can prevent a duplication of efforts by mapping which software that is being preserved and which that are not, and secondly, it can offer metadata and documentation for each piece of software in the database. (Ibid., 3, 11) This would help to organize the work needed to tackle the issues of preserving software. As there is currently no way of distributing the work effectively across institutions and nations, it leads to a situation where ten archivists may be putting their efforts into preserving the same piece of software. (Ibid., 3) A database would thus benefit the organization of the efforts, and tasks could be distributed according to skill-sets. (Ibid., 3) If a collector possesses an original CD-ROM of some rare software, she could make an entry in the database and perhaps image it. Another person could then add documentation and metadata to that entry, and finally, an archivist could take the image and provide an emulation for it. Setting up a registry is thus important as it creates transparency. As Chan (Ibid., 3) concludes, “If you don’t know the status, then you don’t know the scope.”

### *3.2.3 Challenges and limitations*

#### **Funding**

Even within a large institution such as Stanford University, there is still a problem of getting enough funding. Not having enough money is a serious issue as it makes all of the other problems worse. First of all, it limits their ability to recover data from faulty storage media. SUL recently recovered a collection that was stored on six different hard drives. Of the six drives, only two would power. (Ibid., 1) In order to regain access to the data, they had to hire a recovery company that charged them 2500 dollars per drive. In the end, the final cost of restoring the collection totaled at around 10 000 dollars — and even then, they were not able to recover everything. (Ibid., 1) Due to the importance of the collection, the library management agreed to spend this amount of money. However, as Chan notes (Ibid., 1), the amount of efforts put into preserving complex digital objects may come to depend on money. Certain collections or individual pieces of software that are not

considered important enough may be deprioritized. In turn, postponing the migration of software may lead to a loss of this content, as it may not be recoverable in the future.

As the funding usually comes with a deadline, specialized staff is often hired for the duration of a specific project. For both the Meretzky and the Cabrinety collection, Special Collections received extra funding to hire a project archivist for dealing with copyright issues. (Ibid., 5) She would send letters to the copyright owners, asking them for permission to use their work. However, as the project exceeded the funded timeframe, and she had to leave, it reduced their ability to work on settling these kinds of issues. (Ibid., 5) It has now put them in a situation where they are mostly following up on the work that has already been done. As Chan (Ibid., 5) states, they are now specifically targeting pieces of software that are of great interest. If for instance, a scholar contacts them and want to look at a certain game, they will prioritize it. If it becomes of academic interest, they will put more effort into getting the permission for that software.

Having money is paramount to digital preservation. The equipment, services and staff needed to preserve software are all high-cost resources. Another important factor is how the money is spent. As of now, institutions around the world are putting effort into preserving software. However, there is little to no coordination between these institutions, something of which can lead to an ineffective use of resources and a duplication of efforts. This is another benefit of having a national or international registry for software preservation, as it can help institutions coordinate their endeavors. A registry would simplify the distribution of tasks, which in turn would optimize the overall expenditure of resources — if someone has already done a piece of software, others can spend their money elsewhere. In addition to optimizing each institution's spendings, it would also make for a faster progression towards preserving the world's software.

### **Technical challenges**

Preserving software is technically challenging. As mentioned earlier, one of the biggest issues for SUL is a loss of data. This may happen due to a number of reasons. Firstly, they are dealing with software stored on 30-40-year-old storage media. (Ibid., 1) Getting the hardware to power up is therefore one of the main

challenges, and even if it powers, the media may still be too damaged to fully recover the data. (Ibid., 1) As a consequence, it is difficult to predict the recovery success rate of a collection. Even in the cases of healthy storage media, there are still no guarantees that one will be able to fully capture the content.

As the software in SUL's collections are held on a wide range of storage media, and programmed to run on many different platforms, it is often difficult to recognize the file system compatible with the software. (Ibid., 2) A file system is the method used by an operating system to organize and keep track of the files on a computer. (Kirschenbaum, Ovenden, and Redwine 2010, 15) Operating systems do this in different ways, and this poses a challenge when archives are trying to preserve the data. This is because you need the right file system to be able to read it. (Chan 2016, transcript, 2) While larger file systems like MS-DOS are easy to recognize, other environments are far more challenging. If neither the Kryoflux or the FC5025 controllers can recognize the file system, they do however have the ability to do a bit by bit copy using Kryoflux. (Ibid., 10) So if everything else fails, SUL can still make a 1:1 image of the 1s and 0s on the storage media. The captured raw data can of course not be read, but if someone at a later point develops a piece of software to match the bits, it will be possible to regain access to the content.

Even when the file system is recognizable there is still a problem of installing the software, because the software is always part of a specific environment (Ibid., 2) For instance, many of the games owned by SUL are stored on cassette tapes. (Ibid., 1) Having a hardware setup with a tape drive is not sufficient to preserve these games. As these are analog media, the signal first needs to be converted into bits. Then, you need the appropriate software environment that can read and interpret these bits. Sometimes these environments are easy to reproduce, other times they are not — especially when the software depends on libraries that no longer exist, such as libraries for search engines, file format recognition or some other specialized task. (Ibid., 2) Born-digital games might evade this problem, as they are more standardized and therefore more easily confined, but other forms of digital artistic practices may depend on libraries that are not standard. (Ibid., 2)

An artist may use different kinds of software to create an art piece, and this often causes interdependencies that are not easy to keep track of. As a consequence, when SUL receives a hard drive from an artist, they do not just copy out the



individual works. Instead, they reduce the problem of interdependency by imaging the entire disk along with its system files and file structure. (Ibid., 20) This method certainly has its downsides. While an artwork may be a few mega- or gigabytes in size, the entire hard drive may be several terabytes. (Ibid., 21) Avoiding issues related to interdependency is therefore expensive, as it greatly increases the overall cost of storage.

## Legal issues

In addition to the technical challenges of preserving software, SUL also has to deal with copyright issues. Even though they own collections of physical boxes of software, they do not own the actual rights to the software. Consequently, they try to contact the copyright holders to get the permissions they need. (Ibid., 3) A large institution such as Stanford University needs to be cautious as they are likely to get sued if they overstep any legal boundaries. (Ibid., 4) So far, SUL has had some success in getting the permissions, as many of the copyright holders are interested in having their legacy preserved. (Ibid., 4) However, with thousands of software packages to manage, getting in touch with all of the owners can be an arduous task. For instance, if a company developed a piece of software during the 1980s, the ownership of that title may have switched hands multiple times over the last thirty years. Often businesses are acquired by other firms and this complicates the process of tracking down the current copyright holder of a specific software. (Ibid., 4) Even when they are able to track down the assumed copyright holders, the owners may not have the correct papers stating that they have the right to give SUL the permission. (Ibid., 4)

If SUL is unable to acquire the permission, they still image the software. As they own a boxed copy, they migrate its content to another media and at some point in the future they will also emulate it. (Ibid., 4) Not having the permission from the copyright holder does not really limit their ability to preserve the software. It does, however, put serious constraints on their ability to provide public access to it. (Ibid., 4) As long as they have not acquired the appropriate legal permissions, they can only give people access to the artifacts in SUL's reading room. Then they can argue that it is fair use. (Ibid., 4) Thus, in order to use the software, people will

have to visit the reading room at the Cecil H. Green Library at Stanford. This will, of course, lessen the overall impact of the preserved software, especially if compared to a situation where one could freely download it. (Ibid., 4)

## 3.3 The Internet Archive

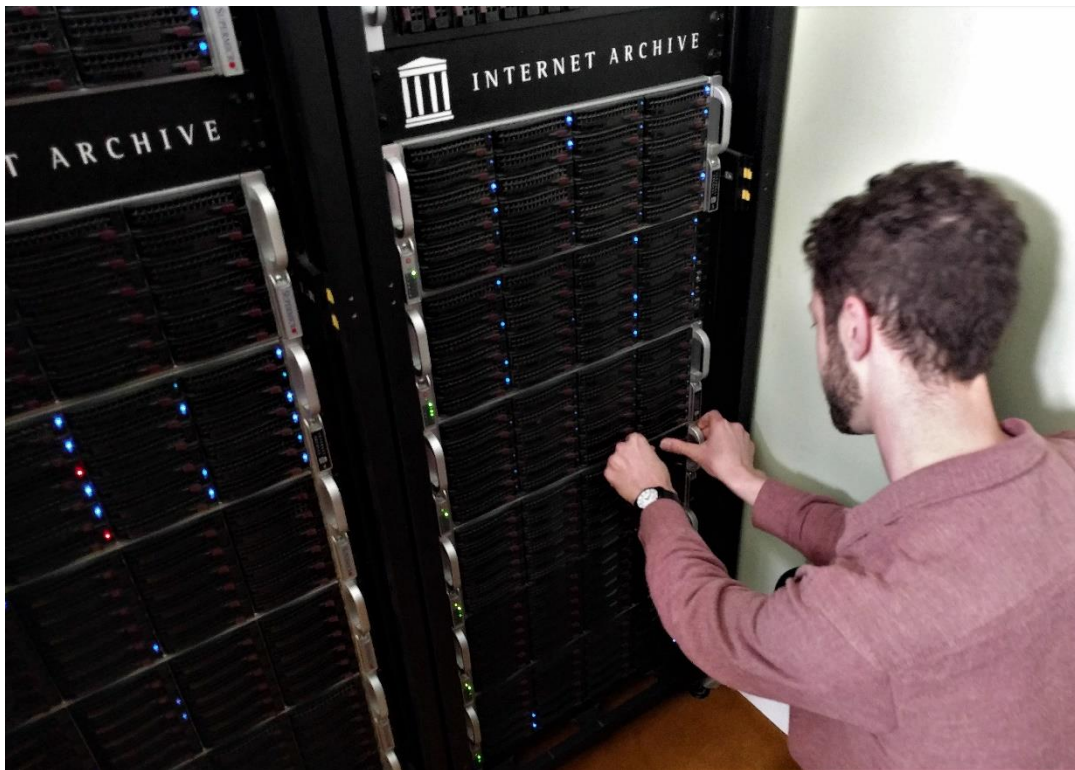
### 3.3.1 *What is the Internet Archive*

The Internet Archive is working to prevent the Internet — a new medium with major historical significance — and other "born-digital" materials from disappearing into the past. (Archive.org 2016a)

The Internet Archive was founded by Brewster Kahle and Bruce Gilliat in 1996 and is located in San Francisco, USA. (Kimpton and Ubois 2006, 202) It is a non-profit organization aiming to collect and preserve the contents of the Internet along with other types of digitally born materials such as texts, audio, moving images and software. The purpose of these efforts is to provide permanent access to historical collections of digital information for researchers, scholars, historians, people with disabilities and to the general public. (Archive.org 2016a)

As of August 2014, the Internet Archive stored 18.5 Petabytes, i.e. 18 500 Terabytes, of unique data. (Archive.org 2016d) This number is probably well over 20 Petabytes in the late of 2016. The data is stored using the Petabox storage system, with a density of 1.4 Petabytes per rack. In their repositories, we find their Wayback Machine, a searchable and browsable collection of fully stored websites, tracking changes in both content and design over time. Further, the archive provides online access to a collection of over 10 million texts, around 300 000 eBooks, a collection of over 1 million images, almost 3 million audio files, 2.5 million movies and close to 125 000 pieces of software. Due to copyright issues, the total amount of data stored on the Internet Archive's servers exceeds the volume of materials that are publically available online. (Kahle 2016, Guided tour

at the Internet Archive) However, at their head office in San Francisco, one can explore the archive in its entirety.



*Image 6: Petabox maintenance at the Internet Archive's head office in San Francisco*

### *3.3.2 Internet Archive's framework for preserving software*

By mid-2016, the Internet Archive preserves and maintain a wide variety of software collections, including the Internet Arcade, the Console Library and the MS-DOS and the Windows 3.x software collections. (Archive.org 2016e) These collections are accessible through the Internet Archive's website and are both browsable and executable. Most of the titles in the collections have their own page containing metadata, documentation, source files and an emulated version of the software.

The Internet Arcade went online in 2014 and “is a web-based library of arcade (coin-operated) video games from the 1970s through to the 1990s...” (Scott 2014; Scott 2016, transcript, 1) The Internet Archive preserves these games and provides public access to emulated versions of the original games. Users can play the games through the Internet Archive's website, and the emulations are able to run on most modern browsers.

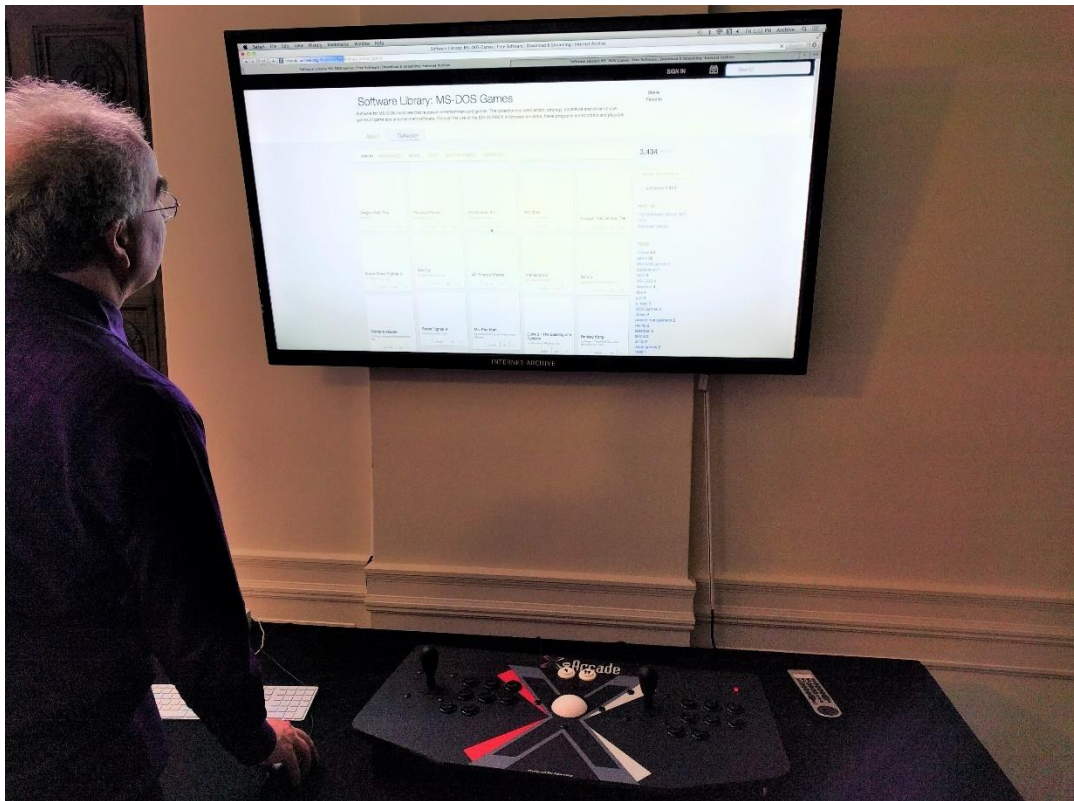


Image 7: Brewster Kahle in the Internet Arcade game room

The Internet Arcade project was initiated by Jason Scott after he was hired in 2011. (Scott 2016, transcript, 1) The Internet Archive had done well with most media types, including books, music, videos and the web, but they had not been successful at archiving software. (Ibid., 1) They did capture a fair amount of software in the early 2000's, but as they lacked a framework for emulation, the project came to a stop. (Ibid., 1) When they resumed their efforts in 2011, Brewster Kahle initially proposed an emulation system similar to bwFLA's *Emulation as a Service* (EaaS). (Ibid., 1) They later realized that this system would not scale properly, meaning that it would not be able to handle a significant increase in traffic. Thus, they came up with a scalable framework using the JSMAME and DOSBox emulators. (Ibid., 1, 25) They have JavaScript ports of these emulators that can receive commands and access disk images through a web-based file system. Simply put, one can hand it a file system with a disk image and then tell it to run the software within a specific browser. (Ibid., 25) IA manages this by using a script that converts these programs into JavaScript objects. (Ibid., 25)

The Internet Archive is not developing their own emulators. Instead, they have built a framework that implements external open-source emulation environments such as the JSMAME and DOSbox. (Ibid., 25) As Scott (Ibid., 44) notes, "...the entire blockchain, the entire chain of our emulation system is open source or uses open source." First of all, this means that their preservation framework does not depend on any proprietary software, and secondly, it means that their framework does not require any development on their side in relation to emulation. The system is set up so that when the developers improve, update or add new features to the emulators, it is automatically transferred. (Ibid., 25) When users start a game, the emulator and program data is loaded into their browser. This is done by using scripts that convert the emulations into browser compatible JavaScript objects. (Ibid., 25)

The Internet Archive does not purchase complete collections of software. Instead, they try to capture every piece of software they can get their hands on, everything from obsolete device drivers to computer games. According to Scott (Ibid., 7), there is little to no deliberate curation involved in the gathering process. Unlike the targeted approach of institutions such as Stanford University Libraries, which acquire software directly from collectors, the Internet Archive collect most of the software themselves. They are ingesting large quanta of materials very quickly, both by grabbing entire FTP-sites of software and by capturing CD-ROMs, DVD-ROMs and floppy disks. (Ibid., 8-9)

The lack of curation and the refusal to differentiate material by its assumed cultural or monetary worth is deliberate. As Scott (Ibid., 11) states, "I don't really play the game of telling people what should and shouldn't be archived." The act of evaluation depends on our ability to recognize value, and one should be very careful about letting people dictate what is important and what is not. (Ibid., 11) The time spent waiting for something to be deemed important is a period in which data can, and most likely will be lost. (Ibid., 7) Being aware of this, the Internet Archive moves somewhat away from the archival concept of appraisal, as they try to curate as little as possible in the gathering process — so as not to color what they collect on the basis of their own presumptions of worth. As a consequence, the Internet Archive has brought attention to previously unrecognized, or even untouched, software and games. (Ibid., 34) Rather than trying to decide what

content is worthy of preservation, they simply move through different media types and gather as much data as possible. (Ibid., 36) So far, they have mostly been capturing large quantities of Windows 5.25 inch floppy disks, Apple II floppy disks and ISO 9660 CD-ROMS. (Ibid., 36) By being attentive to potential institutional biases and the fallibility of value prediction, the Internet Archive has implemented a strategy that strives toward neutrality. Scott remarks;

Well, imagine that nobody collected paintings until 1975, what would you do? You say; “We’ll grab some paintings, let’s classify them.” You look around, and go to every house, every weird place, every attic, every lodge and say; “Hey, how many paintings you’ve got?” “300 paintings” “Hey, can we get those?” (Ibid., 9)

### *3.3.3 Challenges and limitations*

#### **Technical issues**

Regarding software migration, the Internet Archive encounters many of the same problems as Stanford University Libraries. According to Jason Scott (Ibid., 7), their focus is on collecting and capturing as much material as they can, as quickly as possible. He is not too worried about the factor of error involved in the imaging process. (Ibid., 7) Their focus is not on making a perfect copy on the first try, but rather on acquiring and capturing endangered materials as fast as possible and get them onto the archive’s servers. They then reevaluate the scans and do rescans of those that are faulty. (Ibid., 7)

In terms of emulation, the Internet Archive makes up the road as they go. Similar to what the world of computer games calls early-access, the Internet Archive is honest about the fact that they are providing access to emulated games before they have a perfect framework for doing so. Consequently, there will be certain limitations to their emulations. In the blog post, “The MS-DOS Showcase And The Ascension of Version 2”, in which the second version of Internet Archive’s MS-DOS game collection was introduced, Jason Scott (2015) noted that



there were still issues that they had not been able to overcome. Some of the games did not translate well to the browser window, and some of them would crash. In addition, as the games were being streamed from the IA servers, there was no way for the players to save their progress. (Ibid.) On the whole, however, most users would be able to re-experience the era of MS-DOS games within the confines of a browser window. (Ibid.)

Internet speeds and bandwidth usage are also a challenge when it comes to streaming emulated games and software. So far, the Internet Archive has mostly targeted games created during the 1980s and the 1990s, which were usually about 3-4 Megabytes in size. (Scott 2016, transcript, 28) As the Internet Archive transitions to emulating CD-ROMS, it is expected that bandwidth will become a bigger problem. While downloading 3-4 Megabytes of data is barely noticeable in 2016, the same cannot be said about a payload of 400-600 Megabytes. This problem is however expected to be reduced as internet speeds increase. (Ibid., 28)

The Internet Archive does not hang the failure or achievement of their emulation projects on whether or not they accurately recreate the original experience produced by a piece of software. (Ibid., 32) When providing emulated software and games, Scott says, “you are immediately rewarded with complaints about accuracy, fixity, fidelity, context and experience presentation.” (Ibid., 29-30) They might eventually add filters that replicate obsolete TV sets and monitors, but at this point, their focus is on preserving access to the software, rather than recreating the exact experience that they produced. (Ibid., 26-27) They are however experimenting with virtual reality as a way to overcome this problem. One of their team members is working on a VR experience, in which the user will be able to play games on the computers and consoles that they were intended for. (Ibid., 31)

## **Funding**

The Internet Archive is a small non-profit organization that receives funding from government grants and donations from individuals and foundations. (Archive.org 2016a; Archive.org 2016b) As they target a wide variety of digital materials, the limited resources must be distributed across all of their projects, something of which probably involves a certain prioritization of their main

projects, such as the Wayback Machine. The need to prioritize is not really a problem in terms of storage, as the funding that goes into expanding the storage capacity of the archive benefits all of their projects. However, the limited resources do affect the institution's general ability to hire staff, and prevents them from bringing in new people every time they decide to initiate a new project. This was the case with the Internet Arcade. As a general rule, Brewster Kahle will give the go-ahead to any project that does not compromise the integrity of the archive and that can be done on a low budget. (Scott 2016, transcript, 21) Although Kahle supported the Internet Arcade from the beginning, he made it clear that he would not be able to assign a huge amount of employees to the project. (Ibid., 21)

There is a relatively small staff at the Internet Archive who works with software preservation. (Ibid., 1) Although there are many volunteers who have helped to build and maintain the emulation framework, Jason Scott does most of the data-capturing himself. (Ibid., 1, 4) At the time of the interview, he had 2000-3000 CD-ROMS in his office, waiting to be captured. (Ibid., 4) According to Scott, (Ibid., 4) the lack of people working full-time with capturing and migration, has posed a far greater challenge to the project than the issues related to emulation. As a non-profit, the Internet Archive needs to find low-cost approaches to pressing issues within digital preservation. (Ibid., 8) In other words, they need to find clever and inexpensive solutions to complex problems. Although software preservation is considered important by the Internet Archive, it is not prioritized in regard to resources. (Ibid., 21)

On the one hand, the Internet Archive's efforts to preserve software depend on the internal distribution of its resources. On the other hand, it is limited by grant makers that do not want to specifically support software preservation. According to Scott (Ibid., 4), they have been rejected by every grantmaker they have applied to, mostly because it is hard to get across to them why this is a field that needs to be addressed. He ties the difficulty of justifying the need for money to an institutional bias and skepticism toward the cultural value of computer games and software. (Ibid., 4) In this regard, the lack of recognition directly influences the Internet Archive's ability to preserve software, as it restricts the resources they have available. As Scott (Ibid., 4) remarks, they will continue to work on software preservation while they wait for the world to catch up. The continuation of the



Internet Arcade and their other software collections thus depend on the contributions made by committed volunteers who wholeheartedly care about these projects.

## Legal issues

During the early 2000s, the Internet Archive criticized the Digital Millennium Copyright Act and its anti-circumvention regulation, which prohibited any "circumvention of technological measures that effectively control access to copyrighted works". (Archive.org 2016c; Library of Congress Copyright Office 2006) This meant that it was prohibited to jailbreak or hack the encryption technologies controlling the access to copyrighted works. This led the Internet Archive to propose an exemption to the prohibition in 2002, arguing that it crippled archivists' ability to preserve software as it prevented them from migrating the materials from the original and obsolete storage media onto modern storage systems — which in turn would render the software unusable and inaccessible for all time. (Kahle, Lessig, and Seltzer 2002, 13) IA argued that The Digital Millennium Copyright Act blocked the institution's ability to fully archive a significant proportion of their software collection. (Ibid., 2-3) They asserted that digital materials must continuously be transferred to new media before the original media degrades, and be translated to new formats before the original format becomes unreadable by the available technologies. (Ibid, 2-3) Thus they proposed an exemption to the regulation, which would allow libraries and archives, such as the Internet Archive, to copy, migrate and archive software for non-financial purposes — solely to ensure future access to software-based works. (Ibid., 9)

The Internet Archive's proposal was temporarily approved by the Library of Congress Copyright Office in 2003 and approved indefinitely in 2010. (Library of Congress Copyright Office 2006; Library of Congress Copyright Office 2010, 43839) The Library of Congress Copyright Office determined that the prohibition against circumvention of technological measures that effectively control access to copyrighted works should not apply to persons who engage in noninfringing uses of video games and computer programs. (Library of Congress Copyright Office

2010, 43839) The exemption from the anti-circumvention act has set IA free from the legal issues in regards to copying the materials in their collections. As of 2016 they can legally migrate their software to safer storage systems.

In contrast to Stanford University Libraries, the Internet Archive does not stop at migration. They also provide online access to their software collections through their website. Although their tools for emulation are open source, the games and software they are emulating are not necessarily part of the public domain. Most of the software that is made available through the Internet Arcade was made during the 1980s and could thus still be under copyright. US copyright law states that works created after 1978 are under copyright for a duration of 70-120 years after the creator's death. (United States Copyright Office 2011b, 1) Furthermore, § 108(b) and § 108(2) of the U.S Copyright Law states that archives and libraries have the right to migrate and produce digital copies of copyrighted works, as long as the work "that is reproduced in digital format is not otherwise distributed in that format and is not made available to the public in that format outside the premises of the library or archives." (United States Copyright Office 2011a)

Streaming emulated games to the public through a web browser seems to violate this section of the law, but it is difficult to conclude on the legality of their operations before it has been tried in court. The Internet Archive asserts, however, that their use of the software collections is protected by section § 107 of the U.S. copyright law, which states that the fair use of copyrighted works for the purpose of criticism, comment, teaching, research and scholarship are not to be considered infringing use. (Archive.org 2014; United States Copyright Office 2011a) In a blog post titled "A Second Christmas Morning: The Console Living Room", Jason Scott (2013) resonated the stance taken by the Internet Archive on this matter. When announcing the online release of IA's emulated console game collection, he stated that they published the games for the purpose of providing the means for "commentary, education, enjoyment and memory for the history they are part of." (Ibid.)

The Internet Archive may be operating in a legal gray area when it comes to preserving access to software. On the one hand, copyright holders could claim that

the Internet Archive violates intellectual property rights when they provide online access to their collections. On the other hand, the Internet Archive argues that their non-commercial use is noninfringing and protected under fair use. As this is a fairly new field, these cases may not yet have been specifically addressed by the judicial institutions. Until these cases have been tried in court, it is difficult to conclude on whether enabling access to obsolete software is legal or not. The Internet Archive does, however, take measures to protect themselves from being taken to court. Although their initial position is that their use is noninfringing, they state in their copyright policy that they respect intellectual property rights and that they will remove or disable access to content if it appears to infringe the copyright or the intellectual property rights of others. (Archive.org 2014) The game *Street Fighter 2* was for instance taken down after a request by Konami. (Scott 2016, transcript, 33) They will however not remove access to content unless a complaint and proof of ownership are provided by the copyright holders. (Archive.org 2014)

## 3.4 Comparing the institutions

### 3.4.1 Frameworks

As stated in Chapter 2 of this thesis, there are three main strategies for preserving software — hardware preservation, migration and emulation. The methods of the institutions studied in this chapter can be placed within these three categories; the Media Archaeology Lab does hardware preservation, Stanford University Libraries does migration and the Internet Archive does both migration and emulation. In the case of MAL, software is kept alive by sustaining the hardware environment in which the software originally ran. In this way, it can be kept in its original environment with little to no loss in authenticity. What sets the MAL apart from the other two is that the MAL preserves executables indirectly, by ensuring access to the hardware and media needed to read it. In contrast, the SUL and IA preserve information directly through migration.

While the preservation strategy of the MAL is hardware-centric, IA and SUL are information-centric institutions. By this, I mean that their focus is on

sustaining access to information contained by a digital artifact — not the hardware that initially produced that piece of information. By further feeding the migrated data into some emulation software, the IA is also preserving the conceptual aspect of the digital artifact. If we are playing *Out Run* at the Internet Arcade website, we still perceive our experience as being an experience of *Out Run*, even though the Javascript code running in our browser does not resemble that of the original version of the game. The conceptual aspect is therefore kept intact to a degree in which it is perceived as authentic — i.e. the emulation produces a conceptual object that is similar, or equal, to that of the original version of the software. In addition, no changes are being made to the originally migrated code and files, as the emulator software only reads this data and translates it into a browser compatible presentation. In this regard, IA preserves both the authored code and the function of the games in their collections.

The IA preserves software in two stages, first by migrating the code and its dependencies and then by preserving its function through emulation. SUL, however, stops at migration. The folders and files that comprise the software are transferred from the original media and onto safer storage systems. This ensures the longevity of the bits, but it does not preserve the functionality of the software. In this regard, the Internet Archive has a more complete framework for long-term preservation, as they preserve both the static files and the performative function of the software.

Neither SUL nor IA comes close to preserving authenticity in the way that the Media Archaeology Lab does. The MAL allow for a full and authentic experience, that is currently unmatched by emulation. Consider the console game *E.T. the Extra-Terrestrial* running in its original form on an Atari 2600 at the MAL and the browser run emulation of that game provided by the Internet Arcade. While both institutions preserve functional instances of the game, there are some obvious benefits to the MAL version. Although both contain the same narrative, gameplay, rules and graphical elements, they do not share player interaction and controls, nor the overall experience that comes with handling the Atari 2600 hardware. In addition, as mentioned in 3.3.3, the IA's emulators are not perfect — meaning that the outputted presentations can sometimes be glitchy, distorted or in other ways provide inaccurate representations of the original games. In contrast, the MAL preserves every aspect of the game, from the level of inscription and code to the

level of presentation. This makes it possible to analyze the game on all levels, from how the bits of the original game were stored, to how they were processed by the console and finally outputted for presentation. However, although the MAL offers a holistic approach to software preservation, it is not a viable solution for long-term preservation of software. The Internet Archive's strategy of migration and emulation is far less vulnerable to obsolescence and media deterioration, as it does not depend on the existence of spare parts to stay alive.

### *3.4.2 Common challenges*

The overarching problem for all three institutions is a lack of resources. This limits all aspects of their efforts, including their ability to acquire, recover, maintain and expand their collections. It prevents them from hiring specialized staff and it curbs their ability to acquire and maintain the equipment needed to ensure long-term access to their collections. In the case of MAL, the collection is threatened by the fact that they cannot afford to hire a maintenance technician. Lack of funding is also a problem for SUL and IA, as they cannot cover the cost of outsourcing tasks, nor can they hire the staff that they need to acquire, recover and maintain their collections. In addition, short-term funding adds to the problem. As in the case with SUL, the money they receive are often project-oriented grants that come with a deadline. If they cannot complete the project on time, the funding is cut. For SUL, this meant that the capturing process was brought to a halt because they no longer had the money to keep their project archivist. Institutions that deal with long-term software preservation, thus depend on long-term and sufficient funding to properly ensure their collections.

In regard to legal issues, the MAL has the advantage that they can display their collection in their original form on the original hardware. As long as they legally own the hardware and software, they do not infringe on the rights of copyright holders by providing access to it. Furthermore, they do not have to copy or alter the software in order to preserve it. SUL and IA, on the other hand, must make copies of the software to preserve it. Although archival institutions are now allowed to circumvent copyright protection systems to recover software from obsolete media, they do not own the rights to the software they are preserving. This

severely limits their ability to provide access to their collections. While the IA is actively testing the legality by providing access, the SUL is hesitant to do the same out of fear of getting sued. The U.S. copyright laws are not yet adapted to the needs of digital archives. Software preservation institutions are thus exploring uncharted territory and may run the risk of being heavily penalized for their work.

Another issue is the lack of cooperation between the software preservation institutions. Peter Chan (2016, transcript, 3,11) specifically expressed his concerns about the current duplication of efforts. As of 2016, there is no way for archives to know where they should be targeting their efforts. The SUL could, for instance, be migrating a piece of software that has already been migrated by the IA. Conjoining the efforts of information-centric institutions are important because they are all doing the same work. As long as the software is captured correctly, it can be easily transferred between institutions.

## 4. Current situation for software preservation in Norway

### 4.1 Historical outline of digital preservation efforts in Norway

The most prominent archival institutions in Norway are Riksarkivet (The National Archive), which is a subdepartment of Arkivverket (The National Archives of Norway), and Nasjonalbiblioteket (The National Library of Norway). (Kulturdepartementet 2009, 14; Kulturdepartementet 1999, 5) While Riksarkivet is responsible for the preservation of archive material from state institutions, Nasjonalbiblioteket's obligation is to accept all materials made publically available in Norway — including books, films, music albums and the likes. (Arkivverket.no 2016a; Kulturdepartementet 2009, 46)

In 1999, The Norwegian Ministry of Culture (Kulturdepartementet 1999, 4) declared the handling of digitally stored information as one of the main challenges for archives, libraries and museums in the oncoming century. The white paper stated that the frequency of obsolescence in both hardware and software had put electronic materials under serious threat and that there was already a substantial amount of data that had been lost. (Ibid., 5, 41, 66) It stressed that the Norwegian preservation institutions would have to find technical and organizational solutions for preserving a wide range of digital objects, everything from state documents to digital artworks. (Ibid., 20) The ministry thus proposed that Arkivverket should continue to develop methods for acquisition and long-term preservation of electronic archives — containing both digitized representations of non-digital materials and digitally-born artifacts. (Ibid., 5, 20)

In a white paper from 2003, The Norwegian Ministry of Culture reported that a framework for preserving electronic archives had been established and that the tools and methods needed to address these issues were now in place. (Kulturdepartementet 2003, 52) However, what Arkivverket was lacking was the operative capacity to put theory into practice — mainly due to insufficient funding and a lack of staff. (Ibid., 52, 167-168) It was estimated that Arkivverket would need another 30 full-time equivalents for it to properly manage the added stream of electronic materials. (Ibid., 168)

The Committee for Family, Culture and Administration had also expressed their concern for the situation. They argued that the state should provide the resources necessary for Arkivverket to uphold their responsibilities, or else the situation could become unbearable and possibly lead to significant data loss. (Familie- Kultur- og Administrasjonskomiteen 2002, 106) If not taken seriously, Norway could risk losing important fragments of its national heritage. (Kulturdepartementet 2003, 168) The Ministry of Culture, therefore, proposed the initiation of a nationwide project, aimed at reestablishing headroom by sequentially going through the buildup materials from all government agencies and sectors. (Ibid. 168) In addition, it was noted that the efforts to digitize and provide online access to the collections should be strengthened. (Ibid., 169)

The white paper also requested the development of a digital library that would serve as an online access point, unifying the services and information

resources of the country's libraries. (Ibid., 173) A workgroup was appointed in 2002, with the task of outlining an implementation strategy for a digital library. The resulting report was however far more extensive than expected. (Ibid. 173) They argued that such an institution should not be limited to providing access to digital scientific journals and databases. It should also give access to materials that were essential to the Norwegian cultural heritage, including digitized versions of the traditional collections found in the country's archives, libraries and museums. (Ibid. 173) The materials, they continued, should also include non-textual information such as audio recordings, video and other content types. In principle, all kinds of information resources could be gathered (Ibid. 173)

In 2006, Nasjonalbiblioteket established a large-scale digitization program. (Kulturdepartementet 2009, 49) During the 2000s, the infrastructure for digital preservation was established, and a trusted digital repository was built into a mountain in Mo i Rana, just south of the Arctic Circle. (Kulturdepartementet 2003, 53; Kulturdepartementet 2009, 56) The intention of the structure was to ensure the safety of both digitized and digitally-born parts of the country's cultural heritage. (Kulturdepartementet 2003, 53) In a white paper from 2009, the Ministry of Culture announced that Nasjonalbiblioteket now had a flexible and scalable infrastructure for the handling of digital information in place. (Kulturdepartementet 2009, 56) The core of the infrastructure was the repository in Mo i Rana, which at the time had a storage capacity of 4 petabytes. (Ibid. 53) By November 2013, the capacity had increased to 15 petabytes. (Nasjonalbiblioteket 2016b)

In 2009, The Ministry of Culture had stated that the operation of trusted digital repositories should be organized as a cooperative effort between Nasjonalbiblioteket and Riksarkivet. (Kulturdepartementet 2009, 92) As the act of maintaining these facilities were particularly hard on both resources and expertise, the ministry recommended that the responsibilities for the long-term preservation of digital materials should be shared between the two. (Ibid. 92-93)



## 4.2 The status quo and future of Norwegian software preservation

### *4.2.1 Responsibilities and implementation*

Riksarkivet is responsible for the records created by the government's central administration and for the preservation of important archives left by private enterprises or individuals in Norway. In addition to securing these materials, they provide public access to records, both online and through their reading rooms. (Arkivverket.no 2016b; Arkivverket.no 2016a) They have been a part of the forefront of digital preservation in Norway and started accepting deposits of digital materials as early as in the mid-1980's. (Riksarkivet 2010, 5)

In an e-mail message to the author on February 25, 2016, Erik Gunnar Aaberg, Senior Advisor at Riksarkivet, states that they currently have regulations for preserving text, audio, video and map data, but have yet to develop a framework for preserving software. Nevertheless, they do acknowledge the importance and the growing need to establish such a framework. Although they are planning to cooperate with Nasjonalbiblioteket on web preservation, there is no current plan to join their efforts in software preservation. (Aaberg 2016, e-mail message to author, May 6<sup>th</sup>) This is mainly due to limited resources and the need to prioritize other materials. (Ibid.; Aaberg 2016, e-mail message to author, March 30<sup>th</sup>) However, considering the difference in responsibilities between Riksarkivet and Nasjonalbiblioteket, it is more critical that the latter establishes such a framework first. This is because software, especially in the form of art, literature and games, are usually made available to the public through some kind of publication, and thus falls under the responsibilities of Nasjonalbiblioteket.

Nasjonalbiblioteket (NB) is given its mandate through the Norwegian legal deposit law. (Kulturdepartementet 2009, 45) The law requires that all materials that are made available to the public through sale, rental or loan, or in other ways made available outside of the private sphere, must be deposited with NB. (Pliktavleveringslova - avlvl., § 3 ) This is to ensure that the published part of the Norwegian cultural heritage is preserved and documented, and made available for future research and documentation. (Nasjonalbiblioteket 2016a)

Although Norway has gone through a large-span digitization process over the last three decades, no revisions were made to the legal deposit law between its implementation in 1989, and 2016. (Kulturdepartementet 2016a, 4) This led the Ministry of Culture to propose that the law should be updated to meet the challenges of an increasingly digitized society. The proposition was released in 2014 and recommended a revision of the legal deposit law, so that it would apply to all published materials, regardless of its media or form. (Kulturdepartementet 2014, 5) Also digital content publishers and producers would thus be obligated to deposit their material with NB. (Ibid., 5) The proposal was later passed in parliament, and the revisions were implemented January 1<sup>st</sup>, 2016. The adhering regulations to the law will be practiced from January 1st, 2017. (Kulturdepartementet 2016a, 4)

The legal deposit law of 1989 did not address cultural artifacts existing in software form and imposed no requirements for NB to acquire software-based materials. (Ibid., 24) The recent revision changes this. In section § 18 of the revised legal deposit law, it is stated that computer games must be deposited in up to 7 copies, regardless of whether it is a web-based or locally run application. (Ibid., 23) It asserts that only the software that comprises the computer game should be archived — not the environment or the operating system that it operates within. NB must themselves provide the technology needed to keep the game playable, such as gaming consoles, platforms and tools for emulation. (Ibid. 24) Further, the Ministry of Culture notes that they expect contextual materials, such as game-play videos to be publicly available on the web. These materials should, therefore, be gathered by NB through web-harvesting. (Ibid. 24)

Section Manager at NB, Kjersti Rustad writes in an e-mail message to the author on August 18th, 2016, that they are currently accepting deposits of computer games. Due to the previous lack of a legal basis, they did not have software preservation as one of their focal points. As the revised legal deposit law requires them to accept deposits of computer games, they are now establishing a framework for preserving software. However, this field is not entirely new to them, as they have been gathering web-published Norwegian material for several years — including the software found on those websites. These materials have been gathered and preserved according to international standards. In addition, they have been actively preserving single executable files received on media such as CD-

ROMS and external hard-drives. (Rustad 2016, e-mail message to author, August 18<sup>th</sup>)

As of now, NB is at the stage of migration. (Ibid.) In other words, they are not preserving live software. They concentrate their efforts on collecting the software, getting it off of vulnerable storage media and onto safer storage systems. According to Rustad (Ibid.), there are no current plans for providing emulations of the acquired materials. Although progress has been made within the field of emulation, it has yet to mature (Ibid.) It is, however, likely that NB will provide emulations at some point in the future, when the technologies of emulation, virtualization and containment have matured to a point at which only the executable file, its dependencies and its metadata is needed in order to emulate the software. (Ibid.)

#### *4.2.2 Copyright and limitations on access and use*

§ 39h of the current Norwegian copyright law states that a person with the right to use a specific computer program also has the right to copy, alter and adapt that program for the purpose of sustaining functionality or correcting errors. (Åndsverkloven) While the U.S copyright law binds the right of copying and adaptation to the *owner* of a software copy, the Norwegian copyright law binds it to a *right to use*. (United States Copyright Office 2011a, § 117) As discussed in section 1.2.2 of this thesis, the wording of § 117 (a) in the U.S copyright law creates a loophole that software companies can take advantage of. By selling a license to use a computer program rather than selling the actual product, software companies can nullify the right to copy, manipulate and alter the program. Norwegian copyright law appears however not to be affected by software companies' use of EULAs. As long as a person has the right to use a computer program, she also has the right to keep it functional. In addition, § 39i of the Norwegian copyright law further expands this right as it allows for the extraction and translation of a computer program's code, as long as this is done for the purpose of maintaining its compatibility with other software. (Åndsverkloven) Although the copyright law is currently up for consultation and set to be updated,

the contents of § 39h and § 39i will not be changed. (Kulturdepartementet 2016b, 353)

The updated legal deposit law of 2016, will be followed by new revised regulations for both the legal deposit and the copyright law, both of which will take effect on January 1<sup>st</sup> 2017. (Kulturdepartementet 2016a, 36) This is to account for the potential copyright issues related to the use of the digital materials that are to be deposited. (Ibid., 5) The revised regulations related to the copyright law will ensure NB's right to produce copies of materials that have been deposited with them — regardless of whether the materials are protected by copyrights or not. (Ibid., 36) In addition, § 1-10 states that the copies can be produced in other formats than the original format. (Ibid., 36)

In terms of access and use, the proposal states that NB can make the deposited materials available within the confines of its premises, for purposes of research and documentation. (Ibid., 36) Under the previous legislation, digital materials could not be resold or lent by the buyer, without the copyright holder's permission. (Ibid., 8) The new § 1-10a and § 1-10b in the regulation related to the copyright law will, however, allow for further use and access. (Ibid., 8-9) § 1-10a states that researchers, students and employees associated with Norwegian universities or colleges can be given online access to a copyrighted work, as long as it is limited to a maximum of 4 simultaneous users. (Ibid., 36) Users must, however, have a FEIDE ID — an electronic ID used in Norway's educational sector. (Ibid., 13) Further, § 1-10b states that NB can also provide public libraries with access to a copyrighted work if use is restricted to the institutions' premises and limited to a maximum of two simultaneous users. (Ibid., 36) Under no circumstances are users allowed to make digital copies of the copyrighted content. (Ibid., 36)

According to The Ministry of Culture (Ibid., 6), the aim of the new regulations is to ensure that the legal basis for access is adapted to modern times and to the needs of contemporary memory institutions. This involves providing remote digital access to NB's collections. As the consultation document states, the deposited materials should be accessible to the public, regardless of the geographical location or personal economy of the users. (Ibid., 6) In other words, there should be some kind of networked access to the materials. Thus, from 2017,

NB should first and foremost provide digital access to their collections, as long as this can be done within the budgetary and legal frames. (Ibid., 6)

If NB has acquired the appropriate permissions by copyright holders, or if a work is not protected by copyright, these materials can be made publically available on the Internet. (Ibid., 4) Access to collections of copyrighted works can be provided for research and documentation purposes only, as long as it accords with § 1-10, § 1-10a and § 1-10b. Copyrighted works will be stored in a database at NB, and will be accessible through streaming — so as to restrict direct access to the source material. (Ibid., 6-7, 12)

### *4.2.3 Discussion – strengths and weaknesses*

#### **Norwegian law and the legal environment it creates for software preservation**

Regarding acquisition, the updated legal deposit law requires that all works made publically available in Norway should be deposited with NB. As mentioned in 4.2.1, ‘publically available’ is defined as materials that are up for sale, lending or rental or in any other way made available outside of the private sphere.

(Pliktavleveringslova - avlv1., § 3) The last section offers a nearly all-inclusive definition of what should be deposited. Rather than being limited to materials that are made available through the traditional publication channels, the law states that all materials that are made available outside of the private sphere should be deposited. It thus appears that materials published through non-traditional publication channels should also be deposited with NB.

According to the definition offered by the legal deposit law, NB should theoretically own a copy of all software made publically available in Norway. This makes it easier and less costly to build large collections, as it moves both the costs and responsibilities of accumulation from the archive to the content creators and publishers. This may in turn also reduce the risk of institutional bias at the level of acquisition, as the legal deposit law partly removes the need for curation at this level. By that, I mean that the decision of whether or not a work should be acquired is not being based on the library’s own standards of quality or value. Instead, the

materials are collected on the basis of their publication status, as defined by the legal deposit law.

However, using legal deposit as a method for acquisition could be problematic if the law specifically excludes certain materials. In contrast to countries such as Denmark, who included computer games and interactive software in their legal deposit law in 1998, Norway waited until 2016/17 to do the same. (von Suchodoletz, Guttenbrunner, and Rechert 2012, sec. 2.3) Up until now, the software collected and preserved by NB has been given to them voluntarily or gathered in conjunction with their web archiving project. (Rustad 2016, e-mail message to author, August 18<sup>th</sup>) The fact that they have already preserved some amounts of software indicates that the slow progression cannot be tied to a lack of competence or will to preserve software. Rather it is tied to the legislation that governs NB's acquisition process, and the failure of the Norwegian legislature to recognize the importance of preserving executable artifacts. Thus it is likely that a large part of the Norwegian software-based heritage is not being actively preserved by NB. This is especially distressing if we add to the equation that the Danish Royal Library already has a preserved collection of 1500 locally-run computer games. (European Federation of Game Archives Museums and Preservation Projects 2016) Although it may be that Denmark has produced more computer games than Norway over the last 18-19 years, it is still likely that a reasonable amount of Norwegian materials have been neglected or lost during this period.

Although one of the premises for updating the legal deposit law was the emergence of new digital activities such as digital art, computer games, electronic learning resources and virtual exhibitions, the proposition for the law of 2016 and the consultation document describing the new regulations to the copyright and legal deposit law does not address software outside the context of computer games. (Kulturdepartementet 2014, 7; Kulturdepartementet 2016a, 23–24) The reason for this might be that the other materials are considered web-based materials that will be targeted through web-archiving. However, this involves the risk of overlooking software-based digital art and learning resources that are not distributed on the web. There seems to be no explicit obligation to deposit executable works of digital art and literature that are made available through non-orthodox publication channels or exhibited in non-networked spaces such as galleries or conferences. Further, if a learning resource such as the *New Dimensions in Testimony* project, as discussed in

section 2.1.3, was created in Norway, it is not clear whether or not it would be covered by the legal deposit law — it is not a computer game, nor is it made publically available online.

The updated legal deposit law does state in general terms that all materials that are made publically available are to be deposited with NB, but neither the law nor the proposition prior to it mentions locally executable artifacts outside the context of computer games. This also holds true for the consultation document that outlines the upcoming regulations adhering to both the copyright and the legal deposit law. It is, however, worth mentioning that none of these documents explicitly defines what a computer game is. (Kulturdepartementet 2014, 7, 18, 46; Kulturdepartementet 2016a, 23–24) As I discussed in section 2.1.2, works of digital art and literature may have ludic elements and thus intersect with the genre of computer games. Decisions as to what should be deposited and preserved thus depend on how these laws and regulations are interpreted. Will artists, authors and other developers of locally run software interpret the law in such a manner that they will deposit their works, or will they assume that the obligation is limited to web-based materials or to a strict definition of computer games?

On the other hand, there is the question of what will be NB's focus in the upcoming years. As Kjersti Rustad notes, they have not targeted executable artifacts specifically — partly because of the legal deposit law, and partly because they have implicitly gathered much of these materials through web harvesting. (Rustad 2016, e-mail message to author, August 18<sup>th</sup>) How these two factors are weighed in the upcoming years may have an impact on the preservation of locally run executables. If they decide to target their efforts on web-harvesting, and regard it as a sufficient method for acquiring software, they run the risk of overlooking software-based heritage that is not published on the web. The future of Norwegian software preservation thus depends on whether the updated legal deposit law will bring new attention to locally executable artifacts, including genres outside computer games, or if NB will limit themselves to this category. It is, however, difficult to predict how the new law and the adhering regulations will be understood and practiced before they have taken full effect.

While the legislation governing acquisition may have certain flaws or unclarity, it appears that the legislation that decides what archives can and cannot do with the acquired materials provide a sufficient basis for preserving software.

The copyright law allows for copying and altering legally acquired computer programs, and it allows for the extraction of its code. These are rights that need to be in place if archives are to counter the problems caused by obsolescence and the limited lifespan of hardware and operating systems. As software is often stored on short-lived media, it is paramount for archives to be able to extract, copy and migrate the data before the medium dies or becomes unreadable by contemporary hardware. In light of the examples presented in Chapter 2, the copyright law appears to allow for the preservation of software as auxiliary, historical and artistic artifacts. Firstly, the copyright law gives archives the means to maintain functionality of software that is prerequisite to the preservation of other digital artifacts. As the code can be extracted and manipulated, archives could, for instance, rewrite a word-processor so that they could regain access to the information contained in its proprietary document formats. Secondly, the ability to extract a computer program's source code is especially important when preserving works in which the code serves an artistic point. This is because it is necessary to preserve the original code if we are to preserve the authenticity, authorship and the artistic properties of the work. Thirdly, the preservation of code is needed to guarantee scientific reproducibility. If we do not preserve the source code of the computer programs used in scientific research, we may lose the ability to re-assess the calculations performed by those programs. Archived source code thus makes it possible to find and correct potential errors in software that may influence research results.

In terms of providing access to the deposited materials, the previous legislation prohibited the lending of copyrighted digital materials. Although the updated regulations related to the copyright law allows for further access, it will be strictly limited to library premises and to the educational sector. NB will in this regard not have the legal grounds to provide public access to copyrighted software online. This will heavily reduce the social impact of their collections. If we add to the equation that the Norwegian duration of copyright is between 50-70 years, as mentioned in section 1.2.2, it is unlikely that we will see any publically available online collections of playable Norwegian computer games during the next five decades. NB will, however, have the legal basis to provide scholars and students with the access needed to study software-based materials. Thus the legal basis ensures future research while it heavily restricts public access.



## **Nasjonalbiblioteket's framework for software preservation, and how it relates to the challenges encountered by the U.S institutions**

As concluded in section 2.3.5, a viable strategy for software preservation involves a combinatory framework that implements migration, emulation and the acquisition of contextual materials. However, NB only implements two of these methods. Firstly, all acquired software is preserved together with the necessary metadata, including information about the hardware and operating system needed to run it. (Rustad 2016, e-mail message to author, August 18<sup>th</sup>) Further, the consultation document concerning the new regulations related to the copyright and legal deposit law, states that paratextual materials will be acquired through web harvesting and other deposited materials. (Kulturdepartementet 2016a, 24)

Secondly, NB's framework for software preservation is focused on migration, which means that the acquired software will be copied from its original media and securely stored in their trusted digital repository. In this way, NB ensures that we do not lose access to the source code and files that constitute the software. Within the terminology of Jeff Rothenberg, as discussed in section 2.3, NB's strategy is clearly a form of transcription. The source code and files belonging to the software are transferred to a specialized storage system so to avoid that it deteriorates along with its medium. As this process does not involve any transformation of the data, it ensures the integrity, authenticity and authorship of the executable artifact.

Although migration is a necessary stage in the preservation of all digital materials, it should not be treated as the final stage of software preservation. When NB migrates the data pertaining to a piece of software, what they are preserving is the ability to access to the bits — not the ability to understand what those bits mean. As the hardware and operating systems needed to process the data become obsolete, NB's software collections will at some point in the future be rendered unfunctional. Although their current framework does prevent a complete loss of information, it does not preserve software as the performative, dynamic and interactive artifacts that they are.

In order to preserve the performative aspect of software, NB will have to incorporate emulation in their preservation framework. The most pressing issue is that the lack of emulation will prevent scholars, students and library visitors from using the software. This partly compromises the purpose of the legal deposit law, namely that the deposited materials should be preserved and made available as source material for future research. (Pliktavleveringslova - avlvtl., § 1) In addition, one of the motivations for updating the regulations to the copyright and legal deposit law was to provide the educational sector with further access to NB's digital collections. (Kulturdepartementet 2016a, 6) However, the current lack of emulation rules out any long-term access to live software. While their framework may provide sufficient access in regard to researchers that wish to study the source code of a given application, it will severely limit scholars who want to study the gameplay or the mechanics of a computer game. Thus NB's framework raises some serious concerns in relation to ensuring academic access to their software collections.

The lack of emulation is however mostly a problem in regards to access and use — not to a risk of loss. However, until emulation is in place, NB cannot claim to be fully preserving software — especially not if compared to institutions such as the Internet Archive, who already has such a framework in place. NB is also lagging behind Stanford University Libraries in this area. While NB has no immediate plans to incorporate emulation in their framework, SUL has already started planning for emulation and are currently evaluating the available platforms.

NB does, however, avoid some of the problems encountered by the U.S institutions discussed in this thesis. As all published materials in Norway are to be deposited with NB, they do not have to buy collections nor do they depend on collections being donated to them. In addition to having broad access to preservation materials, this also means that NB is less dependent on grants and private donations in order to expand their collections. Further, NB is subject to the Norwegian Ministry of Culture and is financed through the state budget. (Kulturdepartementet 2015, 2) As they are given their mandate through the legal deposit law, and thus are required by law to accept and preserve the deposited materials, it becomes the Ministry of Culture's responsibility to make sure that NB has the funds needed to comply with law. This should make the funding less sporadic and help provide an environment suitable for long-term planning. As I

mentioned in section 3.2.3, SUL often receives project or collection-specific funding that comes with a deadline. When the project period is over, the funding ends regardless of whether the goals have been accomplished or not. NB appears to avoid the problems caused by short-term funding as they are required by law to preserve the deposited materials indefinitely, and thus receives consistent funding over the state budget in order to achieve this goal. (Kulturdepartementet 2016a, 33)

Recovery is, however, a costly and risky endeavor that should be avoided if possible. If software creators interpret the updated legal deposit law in such a manner that they deposit their works, NB will, in theory, have access to these materials as soon as they are published. The legal deposit law may thus allow for a preemptive preservation of software, in the sense that NB can theoretically ingest and preserve large amounts of software as they are being published. From 2017 and onwards it is, therefore, possible to immediately migrate published software before the medium, and the hardware and operating systems needed to read it, become obsolete. In this regard, they could avoid the problems encountered by SUL in relation to recovery. As mentioned in section 3.2.3, SUL spent 10 000 dollars on recovering a single collection and still had a failure rate of 50%. It is, therefore, important that NB exploits the opportunity to take preemptive measures to limit future spendings on recovery and to avoid the risk of high failure rates and the subsequent loss of information.

In terms of issues related to copyright, NB both share and avoid some of the legal issues encountered by IA and SUL. Due to the Norwegian legal deposit and copyright law, NB does not have to get the copyright holder's permission in order to preserve their works. Instead, the content creators and publishers are obligated to deposit a given amount of copies of their work when they make it available to the public. However, if NB wishes to provide public access to collections of copyrighted materials, they will have to acquire the copyright holder's permission to do so. Although the process of acquiring materials is different between NB and SUL, both institutions largely share the same legal challenges in terms of providing public access to their software collections. IA also operates within the same legal framework as SUL, but they appear to be operating in a potentially legal gray area in terms of providing public access to their collections. As NB answers to the Norwegian Ministry of Culture, they are however not in a position to challenge the

law. Consequently, NB cannot provide full public access to their copyrighted materials without changes being made to the Norwegian copyright law.

## 5. Conclusion and ending thoughts

### 5.1 Threats, challenges and motivations

I have argued in this thesis that the vulnerability of digital artifacts can be attributed to both internal and external factors. Firstly, the materiality of executable digital artifacts makes them far more vulnerable than analog objects. Not because they are immaterial artifacts, but because they are performative and multilayered entities that depend on a computer to translate them into humanly meaningful information. The semantic gap between machine language and human language is in this regard what makes these artifacts fragile, as our ability to access them is equivocally dependent on a computer intermediary to translate binary information into meaningful information. However, any computer intermediary will not do. In order to successfully translate the bits into the software we interact with on the screen, we must have a computer with the specific hardware and operating system that knows what these bits mean.

External factors such as obsolescence amplify the problems caused by the human-machine dependency. As the hardware and operating systems needed to process and output a digital artifact becomes obsolete and is replaced by new technologies, we risk losing access to it. Short replacement cycles of technology thus reduce the lifespan of digital artifacts and make them more vulnerable to the passage of time. Secondly, strong copyright laws may interfere with memory institutions' ability to preserve digital materials before they are lost to obsolescence. If they are not allowed to extract and copy the software before either the media onto which it is stored or the environment needed to access it becomes obsolete, the artifact will at some point be lost. Thirdly, the short lifespan of digital artifacts puts pressure on memory institutions to ingest and preserve these materials at a faster rate than analog materials. In turn, this makes it paramount that these

institutions recognize the value of digital materials before it is too late to preserve them. A failure to acknowledge the cultural value of software can, I have argued, be attributed to factors such as new media skepticism and biases. In this case, software is threatened not by the memory institution's ability to preserve, but rather their will to preserve. However, new media skepticism and biases may also influence grant makers or other agents that determine what archives can and cannot do. In this regard, biases or new media skepticism may also limit memory institutions' ability to preserve software.

It is important that memory institutions respond to the challenges outlined in this thesis. First of all, a digital culture is expressed through digital media. This means that parts of our cultural heritage will be embodied in digital artifacts — including software. Further, I give some specific reasons as to why software should be preserved. Firstly, digital artists, authors and game developers use software as a medium of expression. In other words, an executable digital artifact can itself be a culturally important work of art, literature or entertainment. Secondly, software are historical documents that are inherently valuable to our cultural memory. They are important for ensuring scientific reproducibility and they are necessary for understanding the technological development and for the future understanding of our contemporary context. Thirdly, the preservation of software may be prerequisite to the preservation of other digital artifacts — meaning that certain simple digital objects cannot easily be separated from its software environment.

The comparative case-studies shows that the Media Archaeology Lab, the Internet Archive and the Stanford University Libraries have met both similar and dissimilar challenges during their efforts to preserve software. While the latter two are what I call information-centric institutions, in the sense that they preserve software separate from its original environment, the former is a hardware-centric institution that preserves software indirectly by preserving the machines needed to run them. However, the overarching challenge for all three institutions is a lack of funding. Internet Archive's software archivist Jason Scott reported that they had great difficulty of getting grants for game preservation, and he claimed that this could be attributed to an institutional reluctance to accept the cultural value of computer games. Further, both SUL and IA have experienced data loss and unsuccessful recoveries when extracting software from obsolete media. Similarly, the MAL is constantly struggling to keep the collection of obsolete machines alive.

The short lifespan of hardware thus creates problems for both information and hardware-centric institutions. However, MAL evades the legal issues encountered by the SUL and the IA as it does not depend on the ability to copy and migrate materials in order to preserve them. In addition, as the collection consists of hardware devices, there is no need for emulation. Thus, the MAL also avoids the legal issues related to providing large-scale public access to copyrighted materials.

## 5.2 A viable strategy for software preservation

Although hardware-centric institutions such as the MAL evades many of the issues encountered by information-centric institutions such as the SUL and the IA, there is one fundamental reason why hardware preservation is not a viable method for long-term preservation of software — namely that it is not possible to sustain obsolete hardware indefinitely. As both the production and service of the hardware are discontinued, it makes it increasingly difficult for hardware-centric institutions to maintain their collections. The cost of spare parts are high, and these costs increase as they become increasingly rare to find — and at some point in time, they will be impossible to find. Lori Emerson also confirmed that there will come a day when the MAL no longer has a collection of working machines.

A viable strategy for software preservation, therefore, depends on separating the data from its original environment. By doing so, it is possible to protect the data from being trapped in an environment of dead and obsolete hardware. Thus a viable strategy includes the method of migration, which means that we copy the data from the original hardware and transfer it to a newer and safer storage system. However, if we stop at this point, we have only protected the data from deteriorating along with the hardware onto which it is stored. We have not protected the performance of the software from the threats of obsolescence. As the hardware and operating systems needed to interpret the data becomes obsolete, we lose access to the outputted presentation. Consequently, we must also implement the method of emulation in our framework. By using an emulation software that simulates obsolete hardware and operating systems it is possible to regain access to the performance of the preserved data. However, emulation may not always reproduce the exact experience of the original. In addition, we may not have access to the

correct peripheral hardware such as the original controllers or presentation devices. Thus we also need to preserve contextual materials that provide information *about* the software.

In conclusion, the findings in this thesis suggest that a viable strategy for long-term preservation of software must include the methods of migration and emulation in combination with the gathering of metadata, documentation and paratexts. In this way, we preserve the authenticity and authorship of the work by still maintaining access to its performative and conceptual aspect. Finally, the contextual materials provide information *about* the software and fill the potential gaps caused by imperfect emulations or other factors that may hinder an exact recreation of the original experience.

## 5.3 Norwegian software preservation

### 5.3.1 Findings and issues

Out of the two largest Norwegian memory institutions that preserve digital materials, this thesis finds that only NB has a framework for software preservation in place. As this institution is given its mandate through the legal deposit law, it may not be a problem that the responsibility of preserving software is mainly given to one single institution. According to the legal deposit law, all materials made publically available in Norway are to be deposited with NB. This all-encompassing definition of what materials should be deposited makes it reasonable to assume that most software will be affected by the law.

This thesis also finds that Norwegian software preservation is still in its early beginnings. This is mainly because software has not been addressed by the previous legal deposit law. However, this has somewhat changed as the law was updated to include computer games in January 2016. We will however not see the full effect of this until the updated regulations to the legal deposit and copyright law take effect in 2017. Thus, it appears that the Norwegian legislature has been slow to recognize the importance of preserving software. There could be technical, economic or logistical reasons for this, but this seems unlikely considering that NB

has already been preserving software that has been voluntarily deposited with them or gathered from the web. In this regard, it appears that the potential institutional bias has been not embedded in the memory institution itself but rather in the legislation that has governed it.

From 2017 and onwards, it appears that the Norwegian legislation will provide a sufficient basis for preserving software. It ensures archives' right to copy and alter computer programs, and it allows for the extraction and manipulation of their source code. However, some potential issues have been found. First of all, the updated legislation only specifically addresses computer games. Whether or not other genres of software should be deposited with NB seems unclear and might come to depend on how software creators and publishers interpret the legal deposit law. This could be a problem in relation to acquisition. If creators and publishers of other kinds of software do not deposit their materials, NB will predominantly receive computer games. This means that the software that does not fall within this category may run the risk of being lost. Furthermore, as long as the efforts to preserve software are limited to a single institution, it is especially important that this does not exclude software on the basis of type or genre. The status quo might, however, pose a risk to the longevity of software that does not fall within the category of computer games or that is not published on the web.

Although the legislation might cause some issues at the level of acquisition, it appears to provide the copyright exemptions needed to properly migrate and preserve the data. However, it does restrict NB's ability to provide full public access to preserved copyrighted materials. In other words, NB will not have the legal grounds to adapt the access model of the Internet Archive. Rather, they will be able to provide two forms of access; 1) remote access to scholars, students and employees in the educational sector, and 2) on-premise access for library visitors. Thus, with the exception of the ability to provide remote access to the educational sector, NB's access model will be somewhat similar to that of Stanford University Libraries. However, the real problem for both these institution in relation to providing access appears to lie in their shared preservation framework rather than the surrounding legislation.

The current software-preservation framework of NB is found to be based upon the method of migration. This means that they preserve an executable artifact by copying the data that comprises it, and then transfer it to their trusted digital



repository. In this way, the data is extracted and preserved separately from the original hardware and software environment in which it originally ran. Once the data is migrated, it is preserved along with the metadata needed to identify and manage the artifact, and other paratextual materials collected from the web.

Although the framework of NB makes sure that software is not completely lost, it does not fully preserve its performative aspect. Preserving the data that comprises the software does not protect its performative aspect from the external force of obsolescence. As the hardware and operating systems needed to execute the preserved data becomes obsolete, the preserved software will eventually be rendered unfunctional. In this regard, NB currently has a framework for the long-term preservation of dead software. The main problem with this, I have argued, is that NB will not be able to provide scholars and students with access to live software. This may not be an issue for a researcher who is mainly interested in studying the code or the data underlying a piece of software, but it will cause problems for scholars who are interested in the code's output. Examples of this could be a game researcher who want to study the gameplay of a certain game, a social scientist who is interested in studying gender roles or the use of violence in games, or it could be a software engineer interested in the design or mechanics of historical software. In other words, there may be many cases in which having access to live software is necessary to do research.

As I have argued, it seems that Nasjonalbiblioteket does not have a framework that allows them to fully uphold § 1 of the legal deposit law, which states that the law's purpose is to ensure that publically available materials are preserved and made available for research. With their current framework, the software collections of NB will be rendered unfunctional when the hardware and operating systems needed to interpret them becomes obsolete. In this regard, they cannot be said to preserve long-term access to software. Also, their current framework will not make use of the expanded rights given to them by the upcoming regulations to the legal deposit and copyright law, which gives them the ability to provide the educational sector with remote digital access to their digital collections. If NB is to provide the educational sector with long-term access to software, they will have to make emulation part of their framework.

In summary, it appears that NB's framework sufficiently preserves the data that comprises the software, along with a broad range of contextual materials that

provide information about its historical and social context. By migrating the data, they avoid many of the challenges encountered by MAL, such as the software deteriorating along with its original hardware environment. However, similarly to SUL, their framework does not preserve the software's performance from the compatibility issues caused by the external force of obsolescence. Even though the data is ensured, the ability to process that data will be lost at some point in time. Consequently, NB should include emulation in their framework, so as to ensure future access to live software — not just access to its data.

This thesis shows that although NB recognizes the importance of emulation, they have no plans to implement it in their framework anytime soon. This was mainly due to a lack of mature emulation platforms. Although this may be true, the findings in the comparative case-study suggest that this technology is already available. The Internet Archive is currently offering emulation through their open-source platform and Stanford University Libraries plans on implementing the Emulation as a Service platform shortly. However, as the former is developed with the intent of emulating arcade games, the EaaS platform may be the most attractive candidate for memory institutions that deal with a wide variety of software types.

It is likely that NB will experience some problems when implementing emulation in their framework. Considering that NB will mostly acquire computer games from 2016/2017 and onward, bandwidth will certainly be an issue in relation to providing web-based access to them. As mentioned in section 3.3.3, the Internet Archive is expecting some problems when they transition to emulating games migrated from CD-ROMS. These titles will mostly have been produced during the 1990s and the 2000s, and their size will be limited to the storage capacity of a CD-ROM. As computer games produced in 2016/2017 can reach the size of 5-20 gigabytes, it will demand excessive amounts of bandwidth to provide web-access to their emulations. Thus it is reasonable to assume that NB will only be able to provide access to lightweight computer games and software during the first decade or two. However, all deposited executables should be made available on the library premises — and as obsolescence removes the ability to run these artifacts natively, NB will have to make emulation part of their framework.

### *5.3.2 Future research*

This thesis reveals that the Norwegian legislature has been slow to recognize the importance of preserving software. Furthermore, it is unclear whether NB intends to make a large-scale effort to recover and preserve software that predates 2017. As a consequence, there could be a significant amount of software that has been lost. It would, therefore, be interesting to see studies that map the extent of which Norwegian software-based heritage has been lost.

Whether or not Norwegian software preservation will become a story of success, depends on how NB and software producers respond to the new legislation of 2016/2017. At this point, it is however difficult to predict how the next years will pan out. Thus it would be beneficial to reassess the situation in coming years when they have fully adapted to the new legislation. Such a study could clarify some of the questions left unanswered by this thesis. Will NB in practice be limited to preserving computer games and web-harvested software, or will they also preserve software that does not fall within these categories? Will they have implemented emulation in their framework? Will they provide the educational sector with access to software, and if so, how? It would be interesting to see studies on how their framework develops over the upcoming years and what challenges they encounter along the way. Although this thesis attempts to predict some potential problems based on what issues other memory institutions have faced, there is no guarantee that these predictions will be correct. Thus there is a need for future research that maps how NB's plans and strategies for software preservation play out in practice.

# Bibliography

## Interviews/personal correspondence

- Aaberg, Erik Gunnar. E-mail message to author, February 25<sup>th</sup>, 2016
- Aaberg, Erik Gunnar. E-mail message to author, March 30<sup>th</sup>, 2016
- Aaberg, Erik Gunnar. E-mail message to author, May 6<sup>th</sup>, 2016
- Chan, Peter. Interview by Thor Baukhol Madsen. Transcript. Cecil H. Green Library, Palo Alto, April 5<sup>th</sup>, 2016. Accessible at: <http://baukhol.com/Research/ChanTranscript.pdf>
- Emerson, Lori. Interview by Thor Baukhol Madsen. Conversation. The Media Archaeology Lab, Denver, April 12<sup>th</sup>, 2016
- Emerson, Lori. E-mail message to author, April 26<sup>th</sup>, 2016
- Emerson, Lori. E-mail message to author, October 12<sup>th</sup>, 2016
- Kahle, Brewster. Guided tour at the Internet Archive headquarters. San Francisco. April 8<sup>th</sup>, 2016
- Rustad, Kari. E-mail message to author, August 18<sup>th</sup>, 2016
- Scott, Jason. E-mail message to author, October 10<sup>th</sup>, 2016
- Scott, Jason. Interview by Thor Baukhol Madsen. Transcript. Skype, April 20<sup>th</sup>, 2016. Accessible at: <http://baukhol.com/Research/ScottTranscript.docx>

## Literature

- Aanstad, Kristine H. 2011. "Rekordstor Interesse for Platen «Mitt Lille Land»." *Verdens Gang*, October 11.  
<http://www.vg.no/rampelys/musikk/terrorangrepet-22-juli/rekordstor-interesse-for-platen-mitt-lille-land/a/10039908/>.
- Ableton. 2016. "Ableton Specific File Types." Accessed October 18.  
<https://help.ableton.com/hc/en-us/articles/209769625-Ableton-specific-file-types>.
- Allen, Alice. 2013. "We Didn't See This Coming: Our Unexpected Roles as Software Archivists and What We Learned at Preserving.exe." In *Preserving.exe: Toward a National Strategy for Software Preservation*, 23–32. National Digital Information Infrastructure and Preservation Program of the Library of Congress.  
[http://www.digitalpreservation.gov/multimedia/documents/PreservingEXE\\_report\\_final101813.pdf](http://www.digitalpreservation.gov/multimedia/documents/PreservingEXE_report_final101813.pdf).
- Andrews, Robert. 2013. "Forget Digital Media; Let's Get Physical." *The Guardian*, October 1. <http://www.theguardian.com/media-network/media-network-blog/2013/oct/01/digital-media-physical-innovation>.
- Andris, Robert P. 2002. "Do Software Manufacturers Still Have the License to Sue?" *Intellectual Property & Technology Law Journal* 14 (10).  
<http://www.rmkb.com/tasks/sites/rmkb/assets/image/28.pdf>.
- Archive.org. 2016a. "About the Internet Archive." Accessed May 28.

- <https://archive.org/about/>.
- . 2016b. “Credits.” Accessed September 28.  
<http://archive.org/about/credits.php>.
- . 2016c. “Internet Archive Gets DMCA Exemption To Help Archive Vintage Software.” Accessed September 29.  
<https://archive.org/about/dmca.php>.
- . 2016d. “Petabox.” Accessed June 7. <https://archive.org/web/petabox.php>.
- . 2016e. “The Internet Archive Software Collection.” Accessed September 26. <https://archive.org/details/software>.
- . 2014. “Internet Archive’s Terms of Use, Privacy Policy, and Copyright Policy.” <https://archive.org/about/terms.php>.
- Arkivverket.no. 2016a. “About Us.” Accessed August 29.  
<http://www.arkivverket.no/eng/content/view/full/817>.
- . 2016b. “Statement of Service.” Accessed August 29.  
<http://www.arkivverket.no/eng/The-National-Archives/About-us/Statement-of-Service>.
- Assmann, Jan. 1995. “Collective Memory and Cultural Identity.” *New German Critique* 65: 125–33. doi:10.2307/488538.
- Azad, Mohammad Mir, Abu Hasnat Shohel Ahmed, and Asadul Alam. 2010. “Digital Rights Management.” *International Journal of Computer Science and Network Security* 10 (11): 24–33.
- Bartle, Richard A. 2014. “Archaeology Versus Anthropology: What Can Truly Be Preserved.” In *Preserving Complex Digital Objects*. London: Facet Publishing.
- Beagrie, Neil. 2003. “National Digital Preservation Initiatives: An Overview of Developments in Australia, France, the Netherlands, and the United Kingdom and of Related International Activity.” Washington D.C.
- Benjamin, Walter. 1936. “The Work of Art in the Age of Its Technological Reproducibility [Second Version].” In *The Work of Art in the Age of Its Technological Reproducibility, and Other Writings on Media*, edited by Michael W. Jennings, Brigid Doherty, and Thomas Y. Levin, 19–56. Cambridge: Harvard University Press.
- Besek, June M. 2003. “Copyright Issues Relevant to the Creation of a Digital Archive: A Preliminary Assessment.” *Council on Library and Information Resources*. Washington D.C. doi:10.1515/MFIR.2003.86.
- Bierend, Doug. 2014. “The Hackers Who Recovered NASA’s Lost Lunar Photos.” *Wired*, April. <http://www.wired.com/2014/04/lost-lunar-photos-recovered-by-great-feats-of-hackerdom-developed-at-a-mcdonalds/>.
- Blanchette, Jean François. 2011. “A Material History of Bits.” *Journal of the American Society for Information Science and Technology* 62: 1042–57. doi:10.1002/asi.21542.
- Boime, Albert. 1969. “The Salon Des Refusés and the Evolution of Modern Art.” *Art Quarterly* 32.
- Cerf, Vint. 2015. “Digital Vellum and the Expansion of the Internet into the Solar System.” Video lecture. *Carnegie Mellon University*.  
<http://www.cmu.edu/silicon-valley/news-events/dls/2015/cerf-news.html>.
- Collin, Jean-Damien, and Valérie Perrin. 2013. “Collecting Digital Art: Reflections and Paradoxes - Ten Years’ Experience at the Espace Multimédia Gantner.” In *Preservation of Digital Art: Theory and Practice*, edited by Bernhard Serexhe, 48–57. Wien: Ambra Verlag.

- Consultative Committee for Space Data Systems. 2012. "Reference Model for an Open Archival Information System (OAIS)." <https://public.ccsds.org/pubs/650x0m2.pdf>.
- Cullen, Charles T., Peter B. Hirtle, David Levy, Clifford A. Lynch, and Jeff Rothenberg. 2000. *Authenticity in a Digital Environment*. Washington D.C: Council on Library and Information Resources.
- Dappert, Angela, Sébastien Peyrard, Carol C H Chou, and Janet Delve. 2013. "Describing and Preserving Digital Object Environments." *New Review of Information Networking* 18 (2): 106–73. doi:10.1080/13614576.2013.842494.
- Delve, Janet, and David Anderson. 2014. "Introduction." In *Preserving Complex Digital Objects*, edited by Janet Delve and David Anderson, XXXV–XLVII. London: Facet Publishing.
- Drucker, Johanna. 2015. "Performative Materiality and Theoretical Approaches to Interface." *DHQ: Digital Humanities Quarterly* 7 (1): 1–19. <http://www.digitalhumanities.org/dhq/vol/7/1/000143/000143.html>.
- Egenfeldt-Nielsen, Simon, Jonas Heide Smith, and Susana Pajares Tosca. 2013. *Understanding Videogames*. 2nded. New York: Routledge.
- "Electron." 2016. *Encyclopedia Britannica*. Accessed January 20. <http://www.britannica.com/science/electron>.
- ESA. 2015. "Essential Facts About the Computer and Video Game Industry." <http://www.theesa.com/wp-content/uploads/2015/04/ESA-Essential-Facts-2015.pdf>.
- European Federation of Game Archives Museums and Preservation Projects. 2016. "The Royal Library (Denmark)." Accessed November 1. <http://www.efgamp.eu/the-royal-library-denmark/>.
- Familie- Kultur- og Administrasjonskomiteen. 2002. "Budsjett-Innst. S. Nr. 2 (2002-2003)." <https://www.stortinget.no/no/Saker-og-publikasjoner/Publikasjoner/Innstillinger/Budsjett/2002-2003/innb-200203-002/>.
- Flores, Leonardo. 2012. "First Screening by bpNichol." *I Love E-Poetry*. <http://iloveepoetry.com/?p=426>.
- Harvey, Ross. 2012. *Preserving Digital Materials*. 2nded. Berlin: Walter de Gruyter Co.
- Hayles, N. Katherine. 2007. "Electronic Literature: What Is It?" <https://eliterature.org/pad/elp.html>.
- Hong, Neil Chue. 2014. "Digital Preservation and Curation: The Danger of Overlooking Software." In *Preserving Complex Digital Objects*, edited by Janet Delve and David Anderson, 111–23. London: Facet Publishing.
- Huber, Hans Dieter. 2013. "From New Media to Old Media: Ambiguous Concepts, Complex Problems, and Open Questions." In *Preservation of Digital Art: Theory and Practice*, edited by Bernhard Serexhe, 137–44. Wien: Ambra Verlag.
- ISFE. 2012. "Videogames in Europe: Consumer Study - Norway." [http://www.isfe.eu/sites/isfe.eu/files/attachments/norway\\_-\\_isfe\\_consumer\\_study.pdf](http://www.isfe.eu/sites/isfe.eu/files/attachments/norway_-_isfe_consumer_study.pdf).
- Kahle, Brewster, Lawrence Lessig, and Wendy Seltzer. 2002. "Re: RM 2002-4 – 17 USC § 1201 Exemptions Notice of Inquiry." Exemptions Notice of Inquiry. <http://www.copyright.gov/1201/2003/comments/025.pdf>.
- Kanellos, Michael. 2003. "Moore's Law to Roll on for Another Decade." *CNET*. <http://www.cnet.com/news/moores-law-to-roll-on-for-another-decade/>.

- Kimpton, Michele, and Jeff Ubois. 2006. "Year-by-Year: From an Archive of the Internet to an Archive on the Internet." In *Web Archiving*, 201–12. Berlin: Springer. doi:10.1007/978-3-540-46332-0\_9.
- Kirschenbaum, Matthew. 2008. *Mechanisms: New Media and the Forensic Imagination*. Cambridge: The MIT Press.
- Kirschenbaum, Matthew, Richard Oviden, and Gabriela Redwine. 2010. "Digital Forensics and Born-Digital Content in Cultural Heritage Collections." *Council on Library and Information Resources* 149.
- Kittler, Friedrich. 1997. "There Is No Software." In *Literature, Media, Information Systems*, edited by John Johnston, 147–56. New York: Routledge.  
[http://www.monoskop.org/images/a/a6/Kittler\\_Friedrich\\_1992\\_There\\_is\\_no\\_Software.pdf](http://www.monoskop.org/images/a/a6/Kittler_Friedrich_1992_There_is_no_Software.pdf).
- Knoespel, Kenneth J., and Jichen Zhu. 2008. "Continuous Materiality: Through a Hierarchy of Computational Codes." *Théorie Littérature Epistémologie* 25: 235–46.
- Konstantelos, Leo. 2014. "Documenting the Context of Software Art Works Through Social Theory: Towards a Vocabulary for Context Classification." In *Preserving Complex Digital Objects*, 235–56. London: Facet Publishing.
- Kulturdepartementet. 1999. "St.meld.nr.22 (1999-2000)."  
<https://www.regjeringen.no/no/dokumenter/stmeld-nr-22-1999-2000-/id192730/>.
- . 2003. "St.meld.nr.48 (2002-2003)."  
<https://www.regjeringen.no/no/dokumenter/stmeld-nr-48-2002-2003-/id432632/>.
- . 2009. "St.meld.nr.24 (2008-2009)."  
<https://www.regjeringen.no/no/dokumenter/stmeld-nr-24-2008-2009-/id555254/>.
- . 2014. "Prop. 106 L - Endringer I Lov Om Avleveringsplikt for Allment Tilgjengelige Dokument (Innsamling Av Digital Dokument M.m.)."  
<https://www.regjeringen.no/contentassets/a660c098687e4e7f84fbd5908d122412/nn-no/pdfs/prp201420150106000dddpdfs.pdf>.
- . 2015. "Statsbudsjettet 2016 - Tildelingsbrev - Nasjonalbiblioteket."  
<https://www.regjeringen.no/contentassets/2d7c30ab2fd441bc9b3540989a3aea0e/nasjonalbiblioteket-tildelingsbrev-2016.pdf>.
- . 2016a. "Høringsnotat: Ny Forskrift Om Avleveringsplikt for Allment Tilgjengelege Dokument Og Endringer I Forskrift Til Åndsverkloven."  
<https://www.regjeringen.no/no/dokumenter/hoering-ny-forskrift-om-avleveringsplikt-for-allment-tilgjengelege-dokument-og-endringer-i-forskrift-til-andsverkloven/id2505296/>.
- . 2016b. "Høringsnotat - Forslag Til Ny Lov Om Opphavsrett Til Åndsverk Mv. (Åndsverkloven)."  
[https://www.regjeringen.no/contentassets/4c75d8945bc44691925253ca9d0971b5/hoeringsnotat\\_andsverklov.pdf](https://www.regjeringen.no/contentassets/4c75d8945bc44691925253ca9d0971b5/hoeringsnotat_andsverklov.pdf).
- Kuny, Terry. 1998. "The Digital Dark Ages? Challenges in the Preservation of Electronic Information." *International Preservation News: A Newsletter of the IFLA Programme on Preservation and Conservation*, no. 17: 8–13.  
 doi:Article.
- Learneroo. 2016. "What Programming Language Should You Learn?"  
<https://www.learneroo.com/modules/9/nodes/620>.
- Leonardi, Paul M. 2010. "Digital Materiality? How Artifacts without Matter,

- Matter.” *First Monday* 15 (6). doi:10.5210/fm.v15i6.3036.
- Lessig, Lawrence. 1999. *Code*. New York: Basic Books.
- Library of Congress Copyright Office. 2006. “Exemption to Prohibition on Circumvention of Copyright Protection Systems for Access Control Technologies.” *Federal Register* 71 (227): 68472–80.  
<http://www.copyright.gov/fedreg/2006/71fr68472.html>.
- . 2010. “Exemption to Prohibition on Circumvention of Copyright Protection Systems for Access Control Technologies.” *Federal Register* 75 (143): 43825–39.
- London, Bernard. 1932. “Ending the Depression through Planned Obsolescence.” *Self-Published*, 1–20.  
<http://babel.hathitrust.org/cgi/pt?id=wu.89097035273;view=1up;seq=4>.
- “Lov Om Opphavsrett Til Åndsverk M.v. (Åndsverkloven).” n.d.  
<https://lovdata.no/dokument/NL/lov/1961-05-12-2>.
- Machin, David, and Usama Suleiman. 2006. “Arab and American Computer War Games: The Influence of a Global Technology on Discourse.” *Critical Discourse Studies* 3 (1): 1–22. doi:10.1080/17405900600591362.
- Maior, H., David Traum, and Paul Debevec. 2012. “New Dimensions in Testimony.” *PastForward*.  
[http://sfi.usc.edu/sites/default/files/docfiles/PastForward\\_Summer2012\\_0.pdf](http://sfi.usc.edu/sites/default/files/docfiles/PastForward_Summer2012_0.pdf).
- Manovich, Lev. 2003. “New Media from Borges to HTML.” In *The New Media Reader*, edited by Noah Wardrip-Fruin and Nick Montfort, 13–25. Cambridge: The MIT Press.
- Maslin, Keith T. 2007. *An Introduction to the Philosophy of Mind*. 2nd ed. Cambridge: Polity Press.
- Matthews, Brian, Arif Shaon, and Esther Conway. 2014. “How Do I Know That I Have Preserved Software.” In *Preserving Complex Digital Objects*, edited by Janet Delve and David Anderson, 125–39. London: Facet Publishing.
- McDonough, Jerome. 2014. “A Tangled Web: Metadata and Problems in Game Preservation.” In *Preserving Complex Digital Objects*, 185–200. London: Facet Publishing.
- McMullan, Thomas. 2015. “How Art Is Making the Data-Driven City More Liveable.” *The Guardian*, June 11.  
<http://www.theguardian.com/technology/2015/jun/10/art-data-digital-city>.
- Meckien, Richard. 2013. “Cultural Memory: The Link between Past, Present, and Future.” *Institute of Advanced Studies of the University of São Paulo*.  
<http://www.iea.usp.br/en/news/cultural-memory-the-link-between-past-present-and-future>.
- Media Archaeology Lab. 2015. “About.” Accessed November 23.  
[http://mediaarchaeologylab.com/?page\\_id=27](http://mediaarchaeologylab.com/?page_id=27).
- Medietilsynet. 2014. “Barn Og Medier.”  
[http://www.medietilsynet.no/globalassets/publikasjoner/2015/rapport\\_barnog\\_medier\\_2014.pdf](http://www.medietilsynet.no/globalassets/publikasjoner/2015/rapport_barnog_medier_2014.pdf).
- Montfort, Nick. 2008. “ppg256-1 (Perl Poetry Generator in 256 Characters).”  
[http://nickm.com/poems/ppg256-1\\_writeup.html](http://nickm.com/poems/ppg256-1_writeup.html).
- Moore, Gordon E. 1965. “Cramming More Components Onto Integrated Circuits.” *Electronics* 38 (8).  
<https://drive.google.com/file/d/0By83v5TWkGjvQkpBcXJKT1IITTA/view>.
- Nasjonalbiblioteket. 2016a. “Legal Deposit.” Accessed August 30.  
<http://www.nb.no/English/About-us/Legal-Deposit>.



- . 2016b. “Nasjonalbibliotekets Magasiner.” Accessed August 31. <http://www.nb.no/Tilbud/Samlingen/Bevaring/Nasjonalbibliotekets-magasiner>.
- National Archives of Australia. 2016. “Obsolescence - A Key Challenge in the Digital Age.” *National Archives of Australia*. Accessed August 26. <http://www.naa.gov.au/records-management/agency/preserve/e-preservation/obsolescence.aspx>.
- National Library of Australia. 2003. *UNESCO Guidelines for the Preservation of Digital Heritage*. <http://unesdoc.unesco.org/images/0013/001300/130071e.pdf>.
- Negroponce, Nicholas. 1995. *Being Digital*. New York: Vintage Books.
- Nochlin, Linda. 1988. *Women, Art and Power*. New York: Harper and Row.
- Nützel, Jürgen, and Anja Beyer. 2006. “How to Increase the Security of Digital Rights Management Systems Without Affecting Consumer’s Security.” In *Emerging Trends in Information and Communication Security*, 368–80. Berlin: Springer.
- Olson, Michael G. 2013. “Stanford Libraries Preserves Historical Software Collection with Federal Agency.” *SUL Digital Library Blog*. <http://library.stanford.edu/blogs/digital-library-blog/2013/06/stanford-libraries-preserves-historical-software-collection-0>.
- Parikka, Jussi. 2015. *A Geology of Media*. Minneapolis: University Of Minnesota Press.
- Paul, George. 2009. *Foundations of Digital Evidence*. Washington: American Bar Association.
- Pearce-Moses, Richard. 2005. *A Glossary of Archival and Records Terminology*. The Society of American Archivists. <http://files.archivists.org/pubs/free/SAA-Glossary-2005.pdf>.
- “Pliktavleveringslova - Avlvt. Lov Om Avleveringsplikt for Allment Tilgjengelege Dokument (Pliktavleveringslova).” n.d. <https://lovdata.no/dokument/NL/lov/1989-06-09-32>.
- Pogue, David. 2013. “I Have to Rent My Software Now? How Does That Work?” *Scientific American*. [https://www.scientificamerican.com/article/pogue-i-have-to-rent-my-software-now-how-does-that-work/?WT.mc\\_id=SA\\_printmag\\_2013-10](https://www.scientificamerican.com/article/pogue-i-have-to-rent-my-software-now-how-does-that-work/?WT.mc_id=SA_printmag_2013-10).
- Rettberg, Scott. 2014. “Electronic Literature.” In *The Johns Hopkins Guide to Digital Media*, 169–74. Baltimore: Johns Hopkins University Press.
- Riksarkivet. 2010. “Digitalt Og Autentisk - Planlegging Av Ny Depotløsning for Arkivverkets Digitalt Skapte Arkivmateriale.”
- Robinson, Frank Neville H. 2016a. “Electricity.” *Encyclopedia Britannica*. Accessed January 20. <http://www.britannica.com/science/electricity>.
- . 2016b. “Magnetism.” *Encyclopedia Britannica*. Accessed January 20. <http://www.britannica.com/science/magnetism>.
- Robinson, Howard. 2003. “Dualism.” *Stanford Encyclopedia of Philosophy*. <http://plato.stanford.edu/archives/win2016/entries/dualism>.
- Roche, Xavier. 1998. “Copying Websites.” In *Web Archiving*, edited by Julien Masanès, 93–115. Berlin: Springer.
- Rosenbush, Steven, and Steven Norton. 2014. “Virtual and Physical Worlds Converging at Faster Pace, Gartner Forecaster Says.” *The Wall Street Journal*, October 7. <http://blogs.wsj.com/cio/2014/10/07/cios-should-capitalize-on-analog-digital-convergence/>.

- Rothenberg, Jeff. 1999. "Ensuring the Longevity of Digital Information." *Council on Library and Information Resources*.  
<https://www.clir.org/pubs/archives/ensuring.pdf>.
- Scott, Jason. 2013. "A Second Christmas Morning: The Console Living Room." *Internet Archive Blogs*. <http://blog.archive.org/2013/12/26/a-second-christmas-morning-the-console-living-room/>.
- . 2014. "About the Internet Arcade."  
<https://archive.org/details/internetarcade&tab=about>.
- . 2015. "The MS-DOS Showcase (And The Ascension of Version 2)."  
<http://ascii.textfiles.com/archives/4487>.
- . 2016. "Interview by Thor Baukhol Madsen. April 20th 2016." Transcript.
- Serexhe, Bernhard. 2013. "On System Change in Cultural Memory and the Conservation of Digital Art." In *Preservation of Digital Art: Theory and Practice*, edited by Bernhard Serexhe, 75–84. Wien: Ambra Verlag.
- Simanowski, Roberto. 2010. "Reading Digital Literature." In *Reading Moving Letters Digital Literature in Research and Teaching. A Handbook*, edited by Roberto Simanowski, Jürgen Schäfer, and Peter Gendolla, 15–29. Bielefeld: Transcript Verlag.
- Stanford University Libraries. 2016. "Born-Digital Program." Accessed May 1.  
<https://library.stanford.edu/spc/more-about-us/born-digital-program>.
- The National Institute of Standards and Technology. 2015. "About NIST."  
<https://www.nist.gov/about-nist>.
- Thibodeau, Kenneth. 2002. "Overview of Technological Approaches to Digital Preservation and Challenges in Coming Years." In *The State of Digital Preservation: An International Perspective*, 4–32. doi:10.1353/pla.2003.0087.
- Tribe, Mark, and Reena Jana. 2009. *New Media Art*. Köln: Taschen.
- United States Copyright Office. 2011a. "Copyright Law of the United States." *Circular 92*, 8. <http://copyright.gov/title17/circ92.pdf>.
- . 2011b. "Duration of Copyright." *Circular 15a*.
- . 2012. "Copyright Registration for Computer Programs." *Circular 61*.  
<http://www.copyright.gov/circs/circ61.pdf>.
- Vispo. 2006. "The Javascript Version of First Screening."  
<http://vispo.com/bp/javascriptversion.htm>.
- Visual-Arts-Cork. 2016. "Salon Des Refusés." Accessed September 19.  
<http://www.visual-arts-cork.com/history-of-art/salon-des-refuses.htm>.
- von Suchodoletz, Dirk, Mark Guttenbrunner, and Claus Rechert. 2012. "Towards Practical Emulation Tools and Strategies – State of the Art Research Meets Real-World Requirements." iPres Conference 2012.  
<https://ipres.ischool.utoronto.ca/node/73>.
- White, R. 2008. *How Computers Work*. 9thed. Indianapolis: Que Publishing.
- Wingfield, Nick. 2015. "As Downloads Take Over, a Turning Point for the Video Game Industry." *The New York Times*.  
[http://www.nytimes.com/2015/12/21/technology/as-downloads-take-over-a-turning-point-for-the-video-game-industry.html?\\_r=0](http://www.nytimes.com/2015/12/21/technology/as-downloads-take-over-a-turning-point-for-the-video-game-industry.html?_r=0).
- Zyda, Michael, Alex Mayberry, Casey Wardynski, Russell Shilling, and Margaret Davis. 2003. "The MOVES Institute's America's Army Operations Game." *Proceedings of the 2003 Symposium on Interactive 3D Graphics (I3D) 3*: 219–20. doi:10.1145/641480.641523.

# Appendix

## Interview questions

### *Stanford University Libraries and the Internet Archive*

- 1) What are the limitations in respect to what data you can/cannot capture and preserve?
- 2) Do you archive software (computer games etc.), if so, what are your methods and standards for doing so?
- 3) How do you acquire software?
- 4) Do you migrate the bits – have you incorporated emulation in your framework?
- 5) How do you ensure the longevity of the captured software?
- 6) What are the biggest challenges you have encountered in relation to preserving software?
- 7) Are you experiencing other issues – such as issues related to copyrights? How do you avoid copyright infringement?
- 8) How do you regard the future of our digital software-based cultural heritage? Are digital technologies outpacing our archival strategies?

### *The Media Archaeology Lab*

- 1) What are the motivations for preserving hardware? Is the preservation of software one of them?
- 2) What kinds of hardware do you preserve? And what kinds of software is preserved/kept here?
- 3) What are the main challenges of maintaining a functional lab?
- 4) Are there any machines that have failed and that is now beyond repair? Have you received any machines that you have not been able to get working?
- 5) Do you have manuals and documentation for all of the hardware in the collection?

6). How long do you think the collection will stay functional?

### *Nasjonalbiblioteket and Riksarkivet*

1) Bevarer dere software?

2) Hva slags software mottar dere? Spill, elektronisk litteratur og digital kunst?

3) Er det stor spredning på (fil)formatene dere mottar? Og hvordan behandles disse videre?

4) Leveres software inn via nett eller "fysiske" lagringsmedier?

5) Hvor langt har dere kommet i arbeidet og hva er arkiveringsstrategien så langt?

6) Er det viktig å også bevare softwarens funksjon - altså bevare den i et miljø hvor den er kjørbart?

7) Hvis emulering er en fremtidig arkiveringsstrategi, hvilke emuleringsmiljøer planlegger dere å benytte?

8) Hva er de største tekniske utfordringene dere har per dags dato?

9) Har dere støtt på andre utfordringer, for eksempel i forbindelse med opphavsbeskyttet materiale? Får dere bevilget nok penger?