Introduction to Software Curation and Preservation

Eric Kaltman
CLIR Fellow for Data Curation in the Sciences
Overview of Workshop

1. Why Software Preservation?
2. What is Software Curation?
   a. Scope and Description
   b. Migration
   c. Reproduction and Access
   d. Outreach and Culture
3. Trends
4. Efforts at CMU
Overview of Workshop

1. Why Software Preservation?
2. What is Software Curation?
   a. Scope and Description
   b. Migration
   c. Reproduction and Access
   d. Outreach and Culture
3. Trends
4. Efforts at CMU
What is Software?

Source Code
- Instructions to a computational device to do something

Executable binaries
- Compiled source code that executes on in a target environment

Documentation
- Descriptions of implementation and use
What is Software?

Data encoded to be executed by a specific computational context (‘execution context’)

Execution Context - the collection of hardware and software dependencies required to allow for execution

Hardware Dependencies - CPUs, system architecture, peripherals, displays, etc.

Software Dependencies - Compiled libraries, support programs, APIs, etc.
Why Software Preservation?

History
- Modern society is built on and from interactions with software

Progress
- Modern research, regardless of the field, is tied to software infrastructures

Pedagogy
- Access to legacy software and systems for teaching and critical engagement.

Practicality
- Software at the root of commercial progress, ties to competition and legal issues
History and Reproducibility

Research reproducibility
- Maintaining citable access to previous results, analyzes and data sets
- Shifting research practices toward an archival and curatorial mindset
- Models for sustainable software development

History, legacy and maintenance
- Preserving the historical record, the intellectual history of humanity
- Aligning with a maintenance narrative instead of an innovation one
- Access to the past will be dependent on software for the rest of time

Access to software is also access to the files produced and interpreted by software
Digital Dark Ages

“People think that bits are somehow immortal because somehow they’re this ethereal thing in cyberspace...It could be that the format of those bits, the way in which they are interpreted requires a piece of software to figure out what the bits mean. How they should be presented as an image or a video or how you should interact with it in a spreadsheet, but the software doesn’t exist. What if the operating system that the old software used to run on doesn’t exist anymore? What if the latest software doesn’t know how to read the formats of those complex digital objects? Guess what? That information is gone.

There isn’t a systematic way to ensure the information that we create today will still be usable 100 years from now. That’s why I’m worried about a digital dark ages.  

Vint Cerf, co-inventor of TCP/IP
Ensuring the Longevity of Digital Information

Jeff Rothenberg
Scientific American
1995

Figure 4: A bit stream can represent anything at all
Overview of Workshop

1. Why Software Preservation?
2. What is Software Curation?
   a. Scope and Description
   b. Migration
   c. Reproduction and Access
   d. Outreach and Culture
3. Trends
4. Efforts at CMU
Curation

Scope
- What is historically important?
- What will be important for future research efforts?

Description and Standards
- How do we describe software for future access, study and use?

Migration
- How do we transition and preserve data across time?

Storage and Access
- Where do we put these things?
- How do we find them again?
What is Important?

A medical supply company in Miami had received a delivery of botulin, which was to be processed into Botox and distributed. However, it was misprocessed, and a dangerous concentrate was distributed. The FDA had all of the information needed to identify the recipients, but the information was in a file created with a 2003 version of a popular business software application. The 2004 version available to the FDA could not open the data file. The manufacturer of the software was also unable to supply the relevant version.

It so happened that one of the agents involved in the case was familiar with the NSRL, and had in fact provided software to us earlier in the year. He called, explained the situation, and asked if we had the 2003 version of the software. We did! The agent then arranged for an FDA contact to come to NIST, get the software, and put it on a jet to Miami. The people working the case in Miami were able to install the old version, open the data file, and trace the paths of the botulin.
<table>
<thead>
<tr>
<th>Filename</th>
<th>File Size</th>
<th>Release Date</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>doom0_2.zip</td>
<td>254k</td>
<td>Feb. 4, 1993</td>
<td>alpha version - doom0_2.zip downloaded by HEL from <a href="http://www.doomworld.com/pageofdoom">www.doomworld.com/pageofdoom</a>, 18 July 2008</td>
</tr>
<tr>
<td>doom0_4.zip</td>
<td>950k</td>
<td>Apr. 2, 1993</td>
<td>alpha version</td>
</tr>
<tr>
<td>doom0_5.zip</td>
<td>1264k</td>
<td>May 22, 1993</td>
<td>alpha version</td>
</tr>
<tr>
<td>doom1_0.zip</td>
<td>2113k</td>
<td>Dec. 10, 1993</td>
<td>v0.99 shareware version</td>
</tr>
<tr>
<td>doom1_1.zip</td>
<td>2160k</td>
<td>Dec 16, 1993</td>
<td>v1.1 shareware version</td>
</tr>
<tr>
<td>doom1_2.zip</td>
<td>2203k</td>
<td>Feb 17, 1994</td>
<td>shareware version</td>
</tr>
<tr>
<td>doom14bt.zip</td>
<td>2246k</td>
<td>Jun. 28, 1994</td>
<td>v1.4beta shareware version</td>
</tr>
<tr>
<td>doom15bt.zip</td>
<td>2262k</td>
<td>Jul. 8, 1994</td>
<td>v1.5beta shareware version</td>
</tr>
<tr>
<td>doom16bt.zip</td>
<td>2234k</td>
<td>Aug. 3, 1994</td>
<td>v1.6beta shareware version</td>
</tr>
<tr>
<td>dm1666sw.zip</td>
<td>2293k</td>
<td>Sep. 1, 1994</td>
<td>v1.666 shareware</td>
</tr>
<tr>
<td>doom_v18.zip</td>
<td>2423k</td>
<td>Jan 23, 1995</td>
<td>v1.8 shareware</td>
</tr>
<tr>
<td>doom19s.zip</td>
<td>2393k</td>
<td>N/A</td>
<td>v1.9 shareware version</td>
</tr>
</tbody>
</table>
The Oregon Trail

You may:
1. Travel the trail
2. Learn about the trail
3. See the Oregon Top Ten
4. Turn sound off

What is your choice?
The Oregon Trail

Introduction  Options  Quit  Travel the Trail
Overview of Workshop

1. Why Software Preservation?
2. What is Software Curation?
   a. Scope and Description
   b. Migration
   c. Reproduction and Access
   d. Outreach and Culture
3. Trends
4. Efforts at CMU
Migration

Imaging and Storage

Digital Forensic Workflows

Digital Rights Management

Digital Migration Strategies
Migration
CIVILIZATION QUIZ

A usurper claims that you are not the rightful king! To prove you are of royal birth you must answer this question:

Which civilization advances are required to learn the advance pictured below:
- Trade and The Republic
- Masonry and Currency
- Gunpowder and Chemistry
- Writing and Code of Laws
- Philosophy and Trade
- MapMaking and Astronomy
- Computers and Rocketry
- Ceremonial Burial and Code of Laws
- Code of Laws and Literacy
- Currency and Code of Laws

Page 121-130

OK
Digital Migration Strategies

Once imaged or ingested, need a *forever* storage strategy

- Repositories change and evolve, data needs to migrate with those changes
- Provenance information becomes more important

Current repositories are still playing significant catch up
No current long-term solutions available for software data
Overview of Workshop

1. Why Software Preservation?
2. What is Software Curation?
   a. Scope and Description
   b. Migration
   c. Reproduction and Access
   d. Outreach and Culture
3. Trends
4. Efforts at CMU
Reproduction Strategies

Virtualization

Containerization

Emulation

Hardware Preservation
Client Side Emulation

Local executable emulation

JavaScript in browser emulation

Pros:
- Lower latency
- Locally inspectable
- If browser based, shareable and single requirement

Cons:
- Dependent on specific system configuration
- Not easily shareable if local executable
- Legally dubious
Server Side Emulation

Emulation in cloud

Pros:
- Legally more appealing
- Management and maintenance are centralized
- More easy to roll into services

Cons:
- Latency issues
- Much less introspection
- Cloud is a preservation issue in of itself
Hardware Preservation

Cultural software objects are designed for specific hardware and interactions
- Computer Games
- Interactive Art Installation
- Digital Art
- Other digital media works

Socio-cultural context is not in a VM or container

Hardware peripheral and displays are not replicated
Overview of Workshop

1. Why Software Preservation?
2. What is Software Curation?
   a. Scope and Description
   b. Migration
   c. Reproduction and Access
   d. Outreach and Culture
3. Trends
4. Efforts at CMU
Outreach

Locating significant materials around CMU community

Implement reproducible practices inside research labs and departments

Sustainable software development
Overview of Workshop

1. Why Software Preservation?
2. What is Software Curation?
   a. Scope and Description
   b. Migration
   c. Reproduction and Access
   d. Outreach and Culture
3. Trends
4. Efforts at CMU
Sustainable Software Development

Testing oriented

Well documented

Instrumentation

- Continuous integration
- Package management
- Configuration management
- Version Control
Tools and Organizations

Reproducibility Tools Supporting Software

- Code Ocean - http://codeocean.com
- Occam - http://occam.cs.pitt.edu
- Collective Knowledge - http://cknowledge.org/
- Umbrella - http://ccl.cse.nd.edu/software/umbrella/
- ReproZip - https://www.reprozip.org/

Organizations supporting software preservation

- Software Sustainability Institute (SSI) - UK Organization - http://www.software.ac.uk
- Data and Software Preservation for Open Science (DASPOS) - CERN - http://daspos.org
- Software Preservation Network (SPN) - US Memory Institutions
  - http://www.softwarepreservationnetwork.org
ACM Reproducibility

- Repeatability (Same team, same experimental setup)
- Replicability (Different team, same experimental setup)
- Reproducibility (Different team, different experimental setup)
ACM Badging Levels

Artifacts Available
- Software and data present in publication are available for download and investigation

Artifacts Evaluated - Functional
- Software and data have been audited and validated as working

Artifacts Evaluated - Reusable
- Software and data have been audited by a third party, are functional, and significantly oriented toward reusability through documentation, code / software organization, etc.

Results Replicated
- Software and data have been used to validate results

Results Reproduced
- Different software and data have been used to validate results
Overview of Workshop

1. Why Software Preservation?
2. What is Software Curation?
   a. Scope and Description
   b. Migration
   c. Reproduction and Access
   d. Outreach and Culture
3. Trends
4. Efforts at CMU
Tools and Services at CMU

Kilthub Repository
- Source code
- Research data and software executable binaries

Code Ocean (Beta)
- Targeted for August 2019
- Reproducibility platform

Emulation as a Service (EaaS)
- Currently in research beta
- Distributed containerized execution contexts

Software and Data Carpentries
- Two day courses on basic research support tools like Python, R, and Git

History of Science and Technology at CMU (HOST@CMU)
- Interdisciplinary initiative to locate and celebrate CMU technical history
Jupyter Notebooks

In [1]: from scipy.io import wavfile
rate, x = wavfile.read('test_mono.wav')

In [2]:
import matplotlib.pyplot as plt
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
ax1.plot(x); ax1.set_title('Raw audio signal')
ax2.specgram(x); ax2.set_title('Spectrogram')
plt.show()
Emulation as a Service Infrastructure

Interuniversity Distributed Network of Emulation Nodes

Six partner institutions

Environment Contexts with full software installation

Access to legacy files and objects
Questions/Comments

Eric Kaltman
Data Curation Fellow
ekaltman@andrew.cmu.edu