

Interaktion mit multimedialen digitalen Artefakten im Museumskontext – Ein Praxisbericht

Jens-Martin Loebel¹

Abstract: Museen, Bibliotheken und Archive werden in hohem Maße von den durch die Digitalisierung ausgelösten gesellschaftlichen Umbrüchen und Veränderungsprozessen im Umgang mit Information, Bildung und Kultur beeinflusst. Die angereicherte und annotierte Bereitstellung historischer Software mittels Emulation ermöglicht einen je spezifischen Zugang zu den kulturellen Artefakten und erlaubt gleichzeitig verstärkt interaktive, partizipative und narrative Elemente einzusetzen. Es ergibt sich ein großes Potenzial für neue Formen der Mensch-Computer-Interaktion und individualisierte, explorative Wissensvermittlung. Die Ausführung und Bereitstellung im musealen Bereich stellt darüber hinaus weitere Anforderungen. Für eine multimediale Sonderausstellung im Filmmuseum Frankfurt wurde ein modulares Emulationssystem entwickelt und verschiedene Interfaceoptionen zur Nutzerinteraktion evaluiert. Die Ergebnisse werden zusammenfassend vorgestellt und die Notwendigkeit zum Aufbau einer Wissensdatenbank zu Schnittstellen und Best-Practice-Lösungen verdeutlicht.

Keywords: Langzeitarchivierung, Emulation, Computerspiele, Langzeitbewahrung, Museum, Mensch-Computer-Interaktion

1 Einleitung

Die Nutzung von Emulationstechniken zur Präsentation multimedialer Software in Gedächtnisorganisationen wie Museen und Bibliotheken hat in den vergangenen Jahren stark an Bedeutung gewonnen und rückt zunehmend in den Fokus von Forschung und Praxis. Insbesondere Museen, Bibliotheken und Archive werden in hohem Maße von den durch die Digitalisierung ausgelösten gesellschaftlichen Umbrüchen und Veränderungsprozessen im Umgang mit Information, Bildung und Kultur beeinflusst. Als Gedächtnisorganisationen des kulturellen Erbes übernehmen sie gesellschaftliche und kulturelle Aufgaben, die über eine reine Vermittlung von Kunst und Kultur hinausgehen.

Die angereicherte und annotierte Bereitstellung historischer Software mittels Emulation ermöglicht einen je spezifischen Zugang zu den kulturellen Artefakten und erlaubt gleichzeitig verstärkt interaktive, partizipative und narrative Elemente einzusetzen. Es ergibt sich ein großes Potenzial für neue Formen der Mensch-Computer-Interaktion und individualisierte, explorative Wissensvermittlung.

¹ Universität Bayreuth, Medienwissenschaft, Geschwister-Scholl-Platz 3, 95440 Bayreuth, loebel@uni-bayreuth.de

So führte die Deutsche Nationalbibliothek von 2014-2016 im Rahmen des Projekts „EMiL – Bereitstellung von Multimedia-Objekten durch Emulation“ eine großangelegte Pilotstudie zur automatisierten Bereitstellung von Emulationsumgebungen in den Lesesälen der Bibliothek durch (vgl. [St16]). Das Projekt wurde von der DNB positiv evaluiert und ein dauerhaftes Betriebskonzept entwickelt.²

Gleichzeitig stellt die unüberschaubare Anzahl an proprietären Daten- und Dateiformaten sowie sich verändernde Systemschnittstellen zur Nutzerinteraktion eine große Herausforderung dar. Digitale Daten müssen interpretiert werden, damit sie für Menschen in einer sinnlich wahrnehmbaren Form rezipiert werden können. Es müssen skalierbare Wege gefunden werden, sämtliche Komponenten eines Softwarestacks über Generationen von Rechner- und Softwaresystemen hinweg dauerhaft benutzbar zu halten.

Die Zugänglichmachung komplexer multimedialer digitaler Artefakte wie beispielsweise Computer- und Videospiele birgt dabei zusätzliche Anforderungen aufgrund ihres hohen Grads an Interaktivität und der Vereinigung vieler medialer Elemente, beispielsweise Bild, Ton, Video, 3-D. Zeitgleich rückt für Museen die Erhaltung des ursprünglichen Look-and-feel des Spiels und des Dispositivs aus Soft- und Hardware inklusive der Ein- und Ausgabegeräte in den Vordergrund.

Emulation als Softwaretechnik findet in der Informatik breite Anwendung und ist eine gut untersuchte Technik. Ursprünglich in den 1960er Jahren von IBM entwickelt, um alte Programme auf der neu eingeführten und nicht abwärtskompatiblen 360-Rechner-Architektur auszuführen (vgl. [Tu65]), ermöglichen reine Software-Emulatoren heutzutage, die temporale und technologische Kluft zwischen zwei Rechnersystemen zu überbrücken. Die Zugänglichmachung mittels Emulation ist essenziell, um den Charakter komplexer digitaler Objekte zu bewahren und eine authentische Wiedergabe zu ermöglichen (vgl. [Lo14]).

2 Musealer Kontext

Die Ausführung und Bereitstellung im musealen Bereich stellt darüber hinaus weitere Anforderungen. So muss darauf geachtet werden, dass eine einfache Wartung und Austauschbarkeit einzelner Komponenten sowie eine Robustheit der Hardware unter Dauerbenutzung möglich sind. Bei historischen Systemen ist dies naturgemäß nicht gegeben, da die Verfügbarkeit an Originalteilen stark eingeschränkt sein kann. Es müssen daher Best-Practice-Lösungen gefunden werden, welche eine wartbare Kombination von migrierten und originalen Schnittstellen zur Verfügung stellen. Einerseits darf die historische Aufführungspraxis im Sinne der Wissensvermittlung nicht zu stark verfälscht werden, andererseits müssen Interaktionskonzepte auf den Museumsbetrieb abgestimmt sein (vgl. [Lo14], S. 35).

² Vgl. Projektwebsite unter <http://www.multimedia-emulation.de>.

Für die Sonderausstellung „Film und Games. Ein Wechselspiel“ des Deutschen Filmmuseums Frankfurt am Main³ haben wir ein solches modulares Emulationskonzept entwickelt und getestet.

Die Ausstellung soll eine eingehende vergleichende Auseinandersetzung mit den beiden Medien bieten und aufzeigen, in welchem Tempo sich viele Computerspiele in den vergangenen Jahren in Richtung Film entwickelt haben. Wechselwirkungen zwischen Film und Games sollen erstmals im Ausstellungskontext untersucht und Besuchern vermittelt werden, wie sich Filme und Spiele ästhetisch und thematisch beeinflussen.

Dazu war es notwendig Software aus 40 Jahren Computergeschichte und diversen Hardwaregenerationen mittels Emulation über ein einheitliches Interface zur Verfügung zu stellen. Für die Ausstellung wurden spezielle Stationen entwickelt, an denen Besucher Spielszenen interaktiv spielen können. Neben der Betrachtung einer vergleichenden Filmsequenz ist die direkte Ansteuerung bestimmter Szenen von Spielen verschiedenen Alters und unterschiedlichster Systeme notwendig. Konzept und technische Herausforderungen wurden zunächst in einem Prototyp eruiert und die Ergebnisse übernommen (vgl. [Lo15]).

Dazu war einerseits die Entwicklung eines Auswahlmenüs bzw. Systems erforderlich, das den Anforderungen an Ausstellungsbedingungen Rechnung trägt und sowohl robust als auch einfach wartbar ist. Andererseits wurde eine Möglichkeit benötigt, direkt gewisse Spielsequenzen (*Save States*) menügesteuert anspringen zu können, um so eine individuelle Interaktion zwischen Film, Spiel und Besucher herzustellen, die eine tiefergehende Auseinandersetzung mit den Artefakten zulässt.

Neben den üblichen ortsbedingten Anforderungen an die verwendete Hardware wie Platzbedarf, Geräuschentwicklung, Wärmeabfuhr, lag daher das Augenmerk auf der Entwicklung eines Ladesystems von „Save States“ für diese Spiele, eingebettet in ein Auswahlmenü mit Webservice und Skriptsteuerung.

Grundgedanke des Prototyps war die Entwicklung eines Frameworks, das die Machbarkeit des Konzepts zeigt und eine ausbaufähige Architektur bereitstellt. Der Prototyp richtet sich an die Kuratoren der Ausstellung, die durch ein „Ausprobieren“ des Systems Entscheidungen zur Einbettung in die Ausstellung, zu Anpassungen und zum finalen Design des Systems treffen können. Der Fokus liegt auf dem Schnellstart ohne angepasstes Design bzw. Corporate Identity oder Einbettung in ein lauffähiges Gesamtsystem im Gegensatz zu einem Einsatz als Produktivumgebung. Das gesamte System basiert auf Open-Source-Komponenten und wird kostenlos zur Nachnutzung durch andere Museen zur Verfügung gestellt.⁴

Um die Spielsequenzen zu laden wurden Open-Source-Emulatoren angepasst und in das Skriptsystem des Webservice eingebunden. Ein direktes Laden und Ausführen

³ Vgl. Ausstellungswebsite <http://www.filmundgames.de>.

⁴ Software verfügbar unter <http://bitgilde.de/de/>.

bestimmter, durch die Kuratoren vorgegebener, Speicherstände für Besucher wird somit möglich. Zudem ist es möglich Filmsequenzen mit Spielabläufen zu synchronisieren und ab einem gewissen Zeitpunkt Besuchern zu erlauben, vom Film abweichende Handlungsstränge im Spiel auszuprobieren und so explorativ alternative Narrationspfade zu finden.

Das Spielen wird dabei zeitlich limitiert. Die Dauer wird vom Kurator vorher festgelegt und in das Auswahlmenü einprogrammiert. Nach Ablauf der festgelegten Zeit wird das Spiel beendet und es erscheint wieder das Auswahlmenü. So ist das System von mehreren Besuchern nacheinander benutzbar.

Das System stellt ein HTML5-/JavaScript-Auswahlsystem bereit, welches in einem angepassten Firefox (im Kioskmodus) ausgeführt wird. Durch Auswahl werden von einem Webservice mit Skriptsystem bestimmte Speicherstände geladen und/oder ein Videoplayer mit einer bestimmten Filmsequenz auf einem zweiten Bildschirm gestartet. Nach Ablauf einer festgelegten Zeit werden alle Subsysteme beendet und das Auswahlmenü erscheint erneut.

3 Evaluation der Interfacegestaltung

Für die verschiedenen Spieleinstallationen in der Ausstellung ist es notwendig, die gestarteten Emulatoren bzw. Spiele durch Besucher in einheitlicher Form (und unabhängig vom Spiel) wieder beenden zu können. So soll eine Rückkehr zum Auswahlbildschirm ermöglicht und das System für andere Besucher wieder in den Ausgangszustand versetzt werden. Die gezeigten Spiele sollen so auf die gewünschte Spielszene reduziert und der Besucherfluss gelenkt werden.

Es kristallisierten sich die Konzepte eines durch Besucher zu bedienenden „Buttons“ oder einer zeitbasierten Limitierung des Spiels heraus. Die genaue Ausgestaltung verschiedener Button-Lösungen soll im Folgenden näher im Bezug auf ihre Usability diskutiert werden. Aus unseren Tests haben sich für die Button-Lösung zwei weitere Interaktionsmöglichkeiten ergeben.

Für den Prototyp wurde die beschriebene zeitbasierte Limitierung des Spiels umgesetzt, um ein schnelles Testen der verschiedenen Menüoptionen und Interaktionskonzepte zu ermöglichen. Dies stellte sich technisch als robuste und System- / Spiel-unabhängige Lösung heraus. Die Zeit kann vom Kurator/-in dabei frei bestimmt werden.

Das Konzept hat sich bei Tests als sinnvoll für kurze oder wohldefinierte Spielsequenzen erwiesen. Bei komplexeren Interaktionen oder bei zwischenzeitigem Besucherwechsel erscheint ein fester Abbruch jedoch nicht wünschenswert. Hier wäre alternativ eine Lösung zu bevorzugen, welche die Kontrolle über den Spielabbruch den Besuchern überlässt. Die zeitbasierte Lösung sollte allerdings als Fallback oder in bestimmten Situationen zusätzlich erhalten bleiben.

Es soll ein – wie auch immer gearteter – Knopf zur Verfügung gestellt werden, mit dem das Spiel beendet werden kann. Folgende Interaktionskonzepte haben wir evaluiert:

Die Verwendung von Emulatoren ermöglicht es, die Originalsoftware um einblendbare Metadaten und Zusatzfunktionen anzureichern. Denkt man dieses Konzept weiter, so ließe sich diese Technik verwenden um einen virtuellen Button als Screen-Overlay einzubauen, der sich über dem Spielgeschehen einblendet. Um dies zu realisieren, muss sich ein unabhängiges Overlay über den Spielbildschirm legen. Dies erfordert ein tieferes Eingreifen in den Emulator. Gleichzeitig muss ein Overlay-„Widget“ gefunden werden, das mit dem Webservice kommunizieren kann, um die entsprechende Aktion auszulösen.

Hierbei muss jedoch darauf geachtet werden, dass ein solches Overlay von Besuchern nicht als diegetisches Spielelement wahrgenommen wird, da dies die Interaktion mit dem Spiel zu stark verfälschen würde.

In den auf Ubuntu Linux basierenden Stationen kommt der Window Manager „Compiz“ zum Einsatz. Dieser bietet die Möglichkeit, Fenster in verschiedenen Ebenen (Layer) anzuordnen. Um Fenster über allen anderen anzuzeigen gibt es die so genannten „raise above“- und „full screen“- Optionen. Diese Optionen werden beim systemeigenen Widget-System „Screenlets“ (ähnlich MacOS X Dashboard) ebenfalls verwendet. Als Basis für ein vereinheitlichtes Overlayssystem wurde das „WebframeScreenlet“ verwendet, welches eine Website anzeigen kann. Dieses wurde angepasst, um eine Website des zentralen Glassfish-Applikationsservers verkleinert anzuzeigen. Auf dieser befindet sich ein „zurück“-Pfeil. Durch Klick auf den Pfeil wird ein entsprechender Befehl an den gesendet. Weitere Nutzerinteraktion wurde unterbunden und der Fensterrahmen des Widgets unsichtbar geschaltet. Um zu erreichen, dass sich das Widget über andere Fenster legt, wurde mittels des CompizConfig Session Manager (ccsm) in den normalen Bildschirmaufbau auf Betriebssystem-Ebene eingegriffen, und durch entsprechende Regeln das Widget auf eine höhere Zeichenebene gehoben. Zusätzlich kam das System-Programm „wmctrl“ zum Einsatz, um das Widget im System auf Vollbild-Ebene zu verankern.

Tests mit anderen Emulatoren und Spielen zeigten jedoch, dass eine Spiel-unabhängige Lösung nicht möglich ist, da viele Spiele ihrerseits sehr hardwarenah programmiert sind bzw. in das System eingreifen. Dies gilt besonders für den Vollbildmodus vieler Spiele. Werden z. B. direkt die 3D- oder Hardwarebeschleunigungsfunktionen der Grafikkarte angesprochen, ist es nicht möglich, ein Overlay über den Spielbildschirm zu legen. Eine generische Lösung erscheint daher nicht praktikabel, sondern ist immer vom gewählten Emulator und weiteren Komponenten des verwendeten Softwarestacks abhängig.

Unabhängig von Machbarkeitsproblemen hat sich das Interaktionskonzept eines Overlays selbst als das viel schwerwiegendere Problem herausgestellt. Ein Overlay beeinflusst die Interaktion der Besucher mit dem Spiel, so dass Spielfluss und Immersion erheblich gestört werden. So verdeckt das Overlay Teile des Spiels, mit denen die Besucher unter Umständen interagieren müssen. Auch wird das spielfremde Widget als

störend wahrgenommen, da die Interaktion mit diesem in der Regel nicht dem Interaktionskonzept des Spiels entspricht. Wir raten daher aus technischen, medientechnischen und pragmatischen Gründen von solchen Lösungen ab.

Alternativ wurde die Belegung einer bestimmten Taste auf der Tastatur zur Auslösung der Beenden-Aktion über den Webservice getestet. Die Taste wäre dadurch bei jedem Spiel gleich und unabhängig vom Geschehen auf dem Bildschirm. Wird die Taste betätigt, wird das Video unterbrochen und das Abspielprogramm beendet. Es erscheint wieder der Auswahlbildschirm. Das Verankern einer Taste, so dass ein Aufruf systemweit und Applikations-unabhängig funktioniert, erfordert ebenfalls ein Eingreifen in das Betriebssystem. Es wurde mit den Programmen „xkeybind“ und „Ubuntu Shortcuts“ gearbeitet, um die Funktionstaste mit der Ausführung eines Skripts aus der entwickelten Skript-Engine zu belegen.

Bei weiteren Tests hat sich jedoch gezeigt, dass die systemweite Belegung einer Taste von Spielen überschrieben werden kann. Viele Spiele belegen die Tastaturtasten mit Spielefunktionen neu oder fordern beim Betriebssystem die exklusive Nutzung des Tastaturtreibers an, damit Spieler nicht versehentlich eine Betriebssystemspezifische Funktion auslöst, die den Spielfluss unterbricht. Wenn die Taste vom Spiel selbst zur Steuerung benötigt wird ist es nicht möglich, durch Tastendruck das Spiel zu beenden. Eine generische Lösung ist daher mit dieser Variante ebenfalls nicht pragmatisch möglich.

Des Weiteren kann nicht davon ausgegangen werden, dass an jeder Spielstation im Museum eine Tastatur vorhanden ist oder das Anbringen einer solchen praktikabel ist. Oftmals werden von Spielen nur ein Joystick oder eine Maus zur Interaktion verlangt.^[11] Es soll den Besuchern aber eine einheitliche HCI-Schnittstelle zur Verfügung gestellt werden. Da eine generische technische Lösung nicht möglich scheint, wird von dieser Lösung ebenfalls abgeraten.

3.1 Externe Hardware-Buttons

Ein vom Spielsystem gänzlich unabhängige Möglichkeit bildet die Verwendung eines externen Eingabegerätes / Hardware-Buttons. Dieser kann über eine einfache Schaltung beispielsweise an den USB-Port des Rechners angeschlossen werden. Über entwickeltes System-Programm, kann der Knopfdruck über den USB-Bus abgefragt und ein entsprechendes Ereignis ausgelöst werden. Durch die Verwendung eines spielexternen Kanals ins Systems werden die Limitierungen von diegetischen Interfaces umgangen und es kann eine generische Lösung zur Verfügung gestellt werden. Die Nutzung eines dedizierten externen Buttons wurde positiv evaluiert und bietet ein für Besucher leicht identifizierbares und wiederkehrendes Merkmal.

Für eine derartige Lösung liegen zudem bereits Umsetzungen in anderen Ausstellungen vor. Beispielsweise verwendet das Computerspielmuseum Berlin eine solche Lösung über Hardware-Buttons. Diese sind in der ständigen Ausstellung aus stilistischen

Gründen in einen „Competition Pro Star“ Joystick integriert. Das Funktionsprinzip ist aber das Gleiche. Die Buttons im bzw. des Joysticks werden benutzt um eine Videoauswahl und –steuerung zu realisieren.

Des Weiteren konnten wir mit einem ähnlichen Hardwarekonzept für das Jüdische Museum Berlin (entwickelt als RFID-basierte Hardwareerweiterung für die tragbaren Multimedia-Guides) weitere positive Erfahrungen gesammelt werden (vgl. [LK12]).

4 Ausblick

Neuartige Nutzungsmöglichkeiten digitaler Artefakte im Museumskontext bedingen oftmals dedizierte Nutzerschnittstellen um, um eine vereinheitlichte Interaktion mit den Exponaten zu ermöglichen. So lässt sich das Nutzererleben durch die Kuratoren bei der Ausstellungsplanung gezielt beeinflussen.

Solche Schnittstellen stellen Museen ebenfalls vor neue Herausforderungen bzgl. Wartbarkeit und Usability. Zudem zeigt sich, dass für die komplexe Interaktion mit digitalen Unterhaltungsmedien entwickelte Steuerungssysteme (wie moderne Force-Feedback Joypads) nicht nur gewisse Eingewöhnung bei Besuchern bedürfen, sondern auch ergonomisch auf diese abgestimmt sein sollten. So steht die Nutzbarkeit bestimmter Joypads beispielsweise in direktem Bezug zur Handgröße des Benutzers (vgl. hierzu [BM13]). In einem Ausstellungskontext ist es jedoch nicht bzw. nur bedingt möglich, individualisierte Hardware-schnittstellen zur Verfügung zu stellen.

Obwohl dedizierte Hardware Vorsteile durch neue Interaktionsmöglichkeiten erlaubt, liegen bis jetzt wenig generalisierte Praxisevaluationen vor. Es müssen Best-Practice-Lösungen entwickelt und Steuerungssysteme im Hinblick auf Usability, User Experience und Handhabbarkeit im Museumskontext evaluiert werden. Der Aufbau einer Wissensdatenbank für solche Schnittstellen fehlt bis jetzt und stellt ein Desiderat dar.

Literaturverzeichnis

- [BM13] Brown, M. A.; MacKenzie, I. S.: Evaluating Video Game Controller Usability as Related to User Hand Size. In: Proceedings of the International Conference on Multimedia and Human-Computer Interaction - MHCI 2013, International ASET, Inc., Ottawa, Canada, S. 114.1-114.8. Verfügbar online unter: <http://www.yorku.ca/mack/mhci2013-114.pdf>
- [LK12] Loebel, J.-M.; Kuper, H.-G.: RiB-Kit (RFID-in-a-Box) – Eine mobile RFID-Lösung für neuartige Interaktionskonzepte. In: Sieck, J. (Hrsg.): Kultur und Informatik: Aus der Vergangenheit in die Zukunft, Verlag Werner Hülsbusch (vwh), Boizenburg, 2012, S. 187-196.
- [Lo14] Loebel, J.-M.: Lost in Translation – Leistungsfähigkeit, Einsatz und Grenzen von Emulatoren bei der Langzeitbewahrung digitaler multimedialer Objekte am Beispiel

von Computerspielen. Verlag Werner Hülsbusch (vwh), Glückstadt, 2014. Verfügbar online unter: www.translation-gap.de.

- [Lo15] Loebel, J.-M.: Interaktion mit Games mittels Emulation im musealen Kontext. In: Lenhardt, E.; Rauscher, A.; et al. (Hrsg.): Film und Games: Ein Wechselspiel.: Bertz und Fischer, Berlin, 2015. ISBN: 978-3-86505-241-4, S. 230-233.
- [St16] Steinke, T. et al.: Project EMiL - Emulation of Multimedia Objects. In: iPRES 2016. Proceedings of the 13th International Conference on Digital Preservation. Swiss National Library, Bern, 2016. Handle 11353/10.502812, S. 282f. Verfügbar online unter: https://phaidra.univie.ac.at/detail_object/o:502812
- [Tu65] Stuart G. Tucker: Emulation of large systems. In: Communications of the ACM, Band 8, Nr. 12, 1965, S. 753–761.