

Seminar
Langzeitarchivierung
Emulatoren für die Erhaltung von
Computerspielen

Thomas Wettberg
1070084
22.06.2009

Aufgabensteller:
Prof. Dr. Uwe M. Borghoff
Betreuer:
Dipl.-Inform. Nico Krebs



Institut für Softwaretechnologie
Fakultät für Informatik
Universität der Bundeswehr München

Inhaltsverzeichnis

1	Einleitung	5
1.1	Begriffsdefinition: Emulation	5
1.1.1	Software-Emulation	6
1.1.2	Hardware-Emulation	6
2	Motivation	6
3	Überblick über verschiedene Emulatoren	9
3.1	DOSBox	10
3.2	MAME	10
3.3	MESS	11
3.4	JaC64	12
3.5	JPC	12
4	Eignung von Emulatoren zur Langzeitarchivierung	13
4.1	Erörterung	13
4.2	Ausblick und Portierbarkeit	15
5	Literaturverzeichnis	18

1 Einleitung

Seit ihrer Entstehung erfährt die Computertechnik eine rasante Entwicklung [Hoffmann, 2008]. Auf dem Hardware-Markt werden ständig neue Innovationen angeboten, die dem Anwender viel mehr Leistung bieten. Auf diese Entwicklung reagiert die Computerspiele-Industrie und stellt Spiele her, die meistens jedes neue Computersystem nahezu völlig ausreizen, was aus Spieletests unterschiedlicher Magazine hervorgeht¹. Das führt dazu, dass Spiele, die sich in ihrem Veröffentlichungszeitpunkt bereits nur um wenige Jahre unterscheiden, unvereinbare technische Anforderungen an ein Computersystem stellen, da Spiele nicht nachhaltig für neue Computertechnik (z.B. Mehrkern-Prozessoren² oder Grafikkarten mit mehreren Grafikchips³) entwickelt werden können und neue technische Innovationen deshalb für ältere Computerspiele meistens eine unbekannte Umgebung darstellen. Neue Spiele laufen allein schon auf Grund der hohen Leistungsanforderungen nicht auf älteren Systemen und alte Spiele laufen meist nicht auf neuen Systemen, weil sie eben nicht nachhaltig für neue technische Errungenschaften konzipiert wurden. An dieser Stelle setzt der Trend ein, Programme für ein neues Computersystem zu entwickeln, die ein älteres Computersystem emulieren, um letztendlich doch noch ältere Computerspiele auf einem neuen System ausführen zu können. Diese Programme bezeichnet man allgemein als Emulatoren.

In dieser Seminararbeit möchte ich mich dem Thema der Emulation widmen und das Konzept näher beleuchten. Nach einer kurzen Definition des Begriffs „Emulation“ werde ich das Bestreben, Computerspiele zu erhalten, motivieren. Danach gebe ich einen Überblick über Emulatoren für verschiedene Systeme und werde im Anschluss erörtern, inwiefern sich Emulatoren für die Langzeitarchivierung von Computerspielen eignen und wie sich künftig die Portierbarkeit solcher Emulatoren darstellen wird.

1.1 Begriffsdefinition: Emulation

Eine Emulation (Nachbildung, Nachahmung, von lat. aemulator = Nacheiferer) bildet die originale Umgebung digitaler Objekte nach, um sie auch in anderen Umgebungen weiter nutzen zu können [nestor, 2009]. Man unterscheidet Software-Emulation und Hardware-Emulation.

¹Informationen zu aktuellen Spielen findet man beispielsweise unter: <http://www.gamestar.de/> oder <http://www.pcgames.de/m,home/Startseite/>

²ein Produktbeispiel findet man unter: <http://www.intel.com/cd/products/services/emea/deu/processors/core2quad/333915.htm>

³ein Produktbeispiel finden sie unter: http://game.amd.com/us-en/unlock_radeonhd4870x2.aspx?p=1

1.1.1 Software-Emulation

Man spricht von Software-Emulation, wenn der Emulator aus einem Programm besteht, das auf einem System ein anderes System nachbildet. Dieses Programm muss die Architektur des Systems, auf dem es ausgeführt wird als auch die Architektur des zu emulierenden Systems kennen⁴. Die Hauptaufgabe dieses Emulators besteht nun darin, die Befehle (Operation Code) für das zu emulierende System zu interpretieren und in entsprechend neue Befehle, die das ausführende System versteht, zu übersetzen. Weiterhin verwaltet der Emulator einen Teil der Ressourcen des ausführenden Systems und stellt sie dem emulierten Programm so zur Verfügung, wie es das vom emulierten System erwarten würde. Beispielsweise kann sich das Ansprechen des Arbeitsspeichers des ausführenden Systems vom emulierten System unterscheiden. [Rothenberg, 2000]

1.1.2 Hardware-Emulation

Ein reiner Hardware-Emulator besteht im Grunde aus nichts weiter als einer elektronischen Schaltung. Er setzt sich im Inneren aus meist dem aktuellen Stand der Technik entsprechenden elektronischen Bauteilen zusammen, zeigt sich aber nach außen hin funktionell und mechanisch wie das zu emulierende Bauteil. Beispielsweise kann man mit einer 4-Kern-CPU eine 1-Kern-CPU emulieren, indem man die 4-Kern-CPU in ein Gehäuse baut, das der 1-Kern-CPU entspricht, die gleichen Pins anbringt und intern die 4-Kern-CPU so schaltet, dass sie die Befehle einer 1-Kern-CPU verarbeiten kann. In [Wunderlich et al., 2009] wird beschrieben, wie Hardware mit komplexer Logik häufig durch Hardware-Emulatoren emuliert wird, die aus FPGAs (Field Programmable Gate Arrays⁵, sehr flexible elektronische Bauteile) bestehen.

In dieser Arbeit werden ausschließlich Software-Emulatoren betrachtet.

2 Motivation

Computer- und Videospiele zählen zu den ersten digitalen Artefakten, die nur in digitaler Form existent sind⁶. Daraus resultieren spezielle Anforderungen an die Archivierung dieser Artefakte und im Folgenden wird die Archivierung von Computerspielen an sich motiviert werden.

Bis ins Jahr 1958 reichen die Wurzeln heutiger Computerspiele, als der amerikanischen Physiker *William Higinbotham* am *Brookhaven National Labora-*

⁴vgl. hierzu http://www.multimedia.de/artikel/3195_1.php

⁵weitere Informationen zu FPGAs findet man unter: <http://www.fpga4fun.com>

⁶Quelle: http://www.archimuse.com/publishing/ichim04/2758_HuthLange.pdf

tory das Videospiel „*Tennis for Two*“ entwickelte, das aus einem Analogcomputer⁷ und einem Oszilloskop bestand. Dieses Spiel war der Publikumsmagnet an einem Tag der offenen Tür dieses Labors, fand aber aufgrund seiner Einzigartigkeit keine Verbreitung.⁸

Viel bekannter ist das Spiel „*Spacewar*“, das 1962 von *Stephen Russel* und weiteren Mitarbeitern des *Massachusetts Institute of Technology* veröffentlicht wurde. Dieses Spiel war für den ersten Minicomputer der Welt⁹, dem *PDP-1*, programmiert und fand auf diese Weise weitere Verbreitung. Weil dieses Spiel der Initiator der Spielentwicklung ist, wurde *Spaceware* noch häufig kopiert und für andere Plattformen implementiert.¹⁰

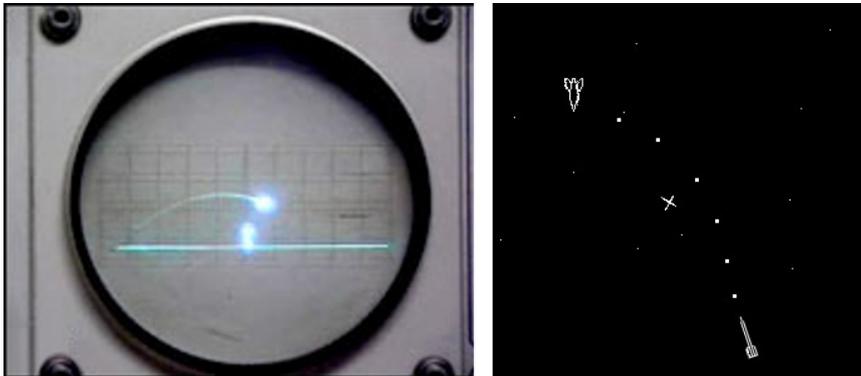


Abbildung 1: Links: „*Tennis for Two*“ Rechts: „*Spacewar*“ Quelle: <http://www.synbiosafe.eu/index.php?page=other-stuff>

Die weit zurückliegenden Erscheinungsjahre dieser ersten Computerspiele zeigen, dass der Beginn dieser Ära fast zwei Generationen¹¹ zurückliegt. Niemand kann mehr bestreiten, dass Computerspiele nicht nur eine kurze Modeerscheinung darstellen, sondern bereits Teil unserer Kultur geworden sind. Man kann davon ausgehen, dass es Eltern gibt, die mit Computerspielen groß geworden sind und ihre Kindheit durch diese digitalen Medien geprägt wurde. So wie sich andere vielleicht in Kindheitserinnerungen, wie dem ersten kompletten Briefmarkensatz oder der ersten eigenen Kamera

⁷Analogrechner unterscheiden sich von Digitalrechnern dadurch, dass sie ihre Werte analog repräsentieren, z.B. durch Längen, Winkel, Wasserständen usw. Weitere Informationen findet man unter: <http://www.analogmuseum.org/>

⁸Quellen: <http://www.osti.gov/accomplishments/videogame.html> und <http://www.bnl.gov/bnlweb/history/higinbotham.asp>

⁹Für die Zeit um das Jahr 1960 war der Computer klein, ist aber mit Heimcomputern aus den 90er Jahren nicht vergleichbar. Er wurde als klein bezeichnet, weil er der erste Computer war, der auch von privaten Endverbrauchern erstanden werden konnte. Weitere Informationen findet man unter: <http://www.dbit.com/~greeng3/pdp1/>

¹⁰Quelle: <http://www3.sympatico.ca/maury/games/space/spacewar.html>

¹¹Nach Karl Mannheim (1893 - 1947), einem Soziologen und Philosoph, erstreckt sich eine Generation über einen Zeitraum von etwa 25 Jahren

schwelgen, erinnern sich gewiss auch Menschen an ihr erstes Computerspiel zurück und möchten ihren Nachfahren davon erzählen und es ihnen am besten vorführen. Selten ist dann noch die alte Rechenmaschine vorhanden, geschweige denn, die Diskette des Spiels in brauchbarem Zustand. Sehr angenehm wäre es, wenn man sich dann nur schnell einen Emulator samt dem Abbild des Spiels aus dem Internet herunterladen müsste und die alten Erinnerungen auf dem eigenen Heimcomputer wieder zum Leben erwecken könnte. Für viele Spiele ist das bereits möglich.

Mittlerer Weile kann man Computerspiele wie Briefmarken, Puppen und Parfums sammeln und es bestätigt nur wieder die Natur des Menschen als „Jäger und Sammler“ oder des Nostalgikers.

Da Computerspiele Bestandteil unserer Kultur geworden sind, findet man sie auch immer häufiger in wissenschaftlichen Arbeiten und Untersuchungen wieder. Vornehmlich in den Bereichen der Psychologie, Pädagogik, Kultur- und Medienwissenschaft werden Computerspiele zitiert, was unter anderem aus folgenden Büchern hervorgeht: [Thomas and Stammermann, 2007] und [Picot et al., 2008]. Ein aktuelles Beispiel für Untersuchungen an Computerspielen ist sicherlich die Diskussion, ob sogenannte „Killerspiele“ die Jugendkultur und die sozialen Kompetenzen der Spieler negativ beeinflussen oder nicht¹². Die Problematik entsteht hier in der Art und Weise der Zitierbarkeit von Computerspielen. Bisher hat sich die „*Digital Games Research Association*“¹³ herausgebildet, die versucht, ein zentrales Register mit Informationen zu allen Computerspielen zu etablieren, das Wissenschaftlern anderer Themenbereiche die Arbeit an und mit Computerspielen dahingehend vereinfachen soll, dass sie Informationen schnell und einfach aus einer von sachkundigem Personal unterhaltenen Datenbank ziehen können. Dieses Register reicht allein natürlich nicht aus. Um wissenschaftliche Arbeiten für die Zukunft überprüfbar zu halten, muss man Spiele langfristig archivieren. Ein weiterer nicht zu vernachlässigender Aspekt dafür, Computerspiele zu erhalten, ist der Nebeneffekt, dass Computerspiele eindrucksvoll dokumentieren, in welchen Etappen sich die Computertechnik entwickelt hat. Ausgerechnet Computerspiele vermögen es, jedes noch so neue Computersystem auszureizen, indem sie virtuelle Welten immer aufwändiger und detaillierter berechnen und grafische Umgebungen kreieren, die der Realität immer näher kommen [Westermann, 2008]. Man behält sich vor zu sagen, Computerspiele würden für Computer gemacht, doch in gewisser Hinsicht ist es nicht zu realitätsfremd zu behaupten, Computer würden für die Spiele entwickelt. Gerade im Bereich der Entwicklung von Grafikkarten ist es auffällig, dass sie darauf getrimmt werden, noch bessere und aufwändigere Effekte zu berechnen, um der Realität noch näher zu kommen (vor allem müssen sie aktu-

¹²nachzulesen unter: <http://www.bundespruefstelle.de/bmfsfj/generator/bpjm/Jugendmedienschutz-Medienerziehung/computer-konsolenspiele,did=108062.html>

¹³weitere Informationen unter: www.digra.org

elle *Shader*- und *DirectX*-Versionen unterstützen¹⁴). Es wird sogar versucht, spezielle Physikbeschleunigungs-Karten¹⁵ auf dem Markt zu etablieren, die vom Computer allein zur Physik-Berechnung¹⁶ virtueller Spielwelten herangezogen werden.

Es ist offensichtlich, dass es gute Gründe gibt, Computerspiele auch für die Zukunft zu erhalten. Es gibt sicherlich noch viele Leute, die sich unter einem Computerspieler einen einsamen, zurückgezogenen und im dunklen Zimmer sitzenden Menschen vorstellen und das Thema mit einem Augenzwinkern belächeln, wenn Computerspielen ein größerer Wert zugesprochen wird. Doch ist das Thema aktueller als man annehmen möchte. Computerspiele sind nunmehr keine Randerscheinung, sondern durchdringen unsere Kultur seit eh und je. Sie sind historische Belege unserer technischen Entwicklung. Sie sind Mittel und Gegenstand wissenschaftlicher Arbeit und sie sind ein Kulturgut unserer Gesellschaft, wie es Schallplatten, Briefmarken und Videokassetten auch sind. Unsere Generation bedauert heute beispielsweise, dass aus der Stummfilmzeit kaum noch Material vorhanden ist. Es wäre ein Fehler, jetzt den Anschluss zu verpassen und Computerspiele nicht zu erhalten. [Huth and Lange, 2004]

3 Überblick über verschiedene Emulatoren

Nachdem die Erhaltung von Computerspielen motiviert wurde, möchte ich nun auf eine Möglichkeit der Erhaltung, nämlich der Emulation, näher eingehen und verschiedene Emulatoren vorstellen. Ich habe eine kleine Auswahl getroffen, die sich nach folgenden Gesichtspunkten unterscheidet. Zum einen sollen diese Emulatoren vielseitig einsetzbar sein und möglichst viele Spiele unterschiedlicher Systeme emulieren. *DOSbox* wurde als bekannter Klassiker aufgenommen und *MAME* als auch *MESS* decken große Bereiche unterschiedlicher Computersysteme ab. *JaC64* ist nur auf ein Computersystem spezialisiert, zeichnet sich aber wie das *JPC*-Projekt durch ausgezeichnete Portierbarkeit aus.

¹⁴Das ist Software, die grafische Effekte berechnet. Weitere Informationen findet man unter: <http://www.gamesforwindows.com/en-US/AboutGFW/Pages/DirectX10-a.aspx>

¹⁵ein Produktbeispiel findet man unter: http://www.nvidia.com/object/nvidia_physx.html

¹⁶Berechnung von Einflüssen der Gravitation, Impulsen, Flugbahnen usw.

3.1 DOSBox

*DOSBox*¹⁷ ist ein Emulator für „*Disk Operating Systems (DOS)*“¹⁸, der sowohl das von *Microsoft* entwickelte *MS-DOS* (1994) als auch das von *IBM* stammende *PC-DOS* emuliert. *DOSBox* ist ein OpenSource-Programm und kann frei aus dem Internet heruntergeladen werden. Es wurde mit der *SDL*-Bibliothek¹⁹, einer plattformübergreifenden Sprachbibliothek, die grundlegenden Zugriff auf Peripherie und Video-/Audio-Hardware ermöglicht, entwickelt und ist deshalb sehr leicht portierbar. Zurzeit gibt es Versionen für folgende Plattformen: *Microsoft Windows*, *FreeBSD*, *Fedora Core (Linux)*, *Gentoo (Linux)*, *Mac OSX*, *OS/2*, *BeOS* und *RiscOS*. Neben der Vielfalt der Plattformen bietet *DOSBox* auch eine Vielfalt an Einstellungen. Sehr wichtig ist beispielsweise die variable Einstellung der Prozessorgeschwindigkeit, da *DOS*-Programme häufig abhängig von der MHz-Zahl unterschiedlich schnell ablaufen und Spiele sonst teilweise unspielbar wären. Weiterhin ist *DOSBox* sehr flexibel einsetzbar. Die heruntergeladenen Dateien sind ohne Installation sofort ausführbar und alle Einstellungen können bei Bedarf in den Textdateien vorgenommen werden. Nachdem *DOSBox* gestartet wurde, öffnet sich ein dem *MS-DOS* ähnlicher Kommandozeileninterpreter, der alle Funktionen der bekannten *DOS*s emuliert. Dazu gehören auch Systemeigenschaften und -funktionen wie *BIOS*²⁰, x86-Prozessor, verschiedene Grafikkartenstandards, CD-ROM-Laufwerk uvm. *DOSBox* ist damit ein sehr mächtiger und vielseitiger Emulator, der fast allen Ansprüchen gerecht wird. Unterstrichen wird dies eindrucksvoll durch die gut 2500 erfolgreichen Spieltests, die auf dem Webauftritt von *DOSBox* eingesehen werden können. Man kann fast davon ausgehen, dass mit *DOSBox* die *DOS*-Spielewelt zum größten Teil erhalten wird. [Kohler, 2005a]

3.2 MAME

*MAME*²¹ ist ein Akronym für „*Multiple Arcade Machine Emulator*“ und ist ein Emulator für Videospiele, die man von den vor allem früher stark verbreiteten Spiele-Automaten kennt. Das besondere an diesen Spielen ist, dass sie vorrangig als Hardware (ROMs) vorlagen und nicht auf mobilen Datenträgern verbreitet wurden. Der Emulator, wieder ein OpenSource-Produkt, bildet die Technik der Spiele-Automaten nach und täuscht Spielen ihre be-

¹⁷Offizieller Internetauftritt: <http://www.dosbox.com/>

¹⁸Informationen zur Entstehung findet man unter: <http://www.computermuseum-muenchen.de/dictionary/history/dos.html>

¹⁹Simple Directmedia Layer, weitere Informationen findet man unter: <http://www.libsdl.org/>

²⁰Basic Input/Output System, ein Programm im Festspeicher eines PCs, das unmittelbar nach dem Start ausgeführt wird und den Start des Betriebssystems einleitet.

²¹Offizieller Internetauftritt: <http://mamedev.org/>

kannte Umgebung mit allen Ressourcen und Funktionen, insbesondere den verschiedenen Eingabegeräten wie Joystick, unterschiedlichen Knöpfen usw. vor. Interessant ist auch, dass Spiele-Automaten häufig mehrere Chips besaßen und der Emulator deshalb mehrere Prozessoren nachahmen muss. Das führt zu hohen Leistungsanforderungen und selbst auf heutigen High-End-Rechnern²² können noch nicht alle Arcade-Spiele auf voller Geschwindigkeit emuliert werden. *MAME* wurde vorrangig für *Windows* entwickelt, konnte bisher aber auf weitere bekannte Plattformen wie *Linux* und *MacOS* portiert werden. Um mit *MAME* ein Spiel zu emulieren, muss der Emulator von einem Kommandozeileninterpreter aus zusammen mit dem Spiel und optionalen Parametern gestartet werden. Die Spiele liegen meist als Abbilder der originalen Hardware vor und wurden in keinster Weise verändert, sodass man vom PC aus auch das Einwerfen von Münzen per Tastatur nachahmen muss. Es gibt etwa 30 verschiedene Ableger von *MAME*, die entweder für andere Plattformen gemacht sind oder eine besondere grafische Oberfläche bieten, die die Benutzung des ursprünglichen Emulators komfortabler gestalten. Bisher konnten geschätzte 5000 Arcade-Spiele erhalten und meist erfolgreich emuliert werden. [Kohler, 2005b]

3.3 MESS

*MESS*²³ steht für „*Multiple Emulator Super System*“ und ist ein Emulator für 252 unterschiedliche Konsolen und Computersysteme (dazu gehören z.B. *C64*, *AMIGA*, *Nintendo64*, *SEGA Mega Drive*, *Atari 7800* uvm.), der bis zu 800 verschiedenartige Ausprägungen dieser Systeme unterstützt. *MESS* nutzt den Kern und vor allem die CPU-Emulation²⁴ von *MAME* und ähnelt diesem Emulator deshalb in vielseitiger Hinsicht. *MESS* hat die gleichen Systemvoraussetzungen wie *MAME* und wurde inoffiziell durch eine Implementierung in *SDL* wie *MAME* auf andere Plattformen als *Windows* portiert. *MESS* ist deshalb komfortabel, da von vornherein neben der Kommandozeilenversion auch eine Ausführung mit grafischer Oberfläche bereitgestellt wird. Die Installation beschränkt sich auf das Entpacken der heruntergeladenen Dateien. Um ein Spiel zu starten, werden zunächst das System und dann das Spiel ausgewählt. In Verbindung mit der grafischen Oberfläche sind Systemeinstellungen leicht vorzunehmen. [Kohler, 2005c]

²²PCs mit der neuesten Technik, die auf dem zivilen Markt erhältlich ist

²³Offizieller Internetauftritt: <http://www.mess.org/>

²⁴Emulatoren bilden intern einzelne Ressourcen nach, die CPU-Emulation ist das Modul, das den Prozessor implementiert

3.4 JaC64

Der Emulator „*JaC64*“ kann sich mit *MAME* und *MESS* nicht messen, weil er nur den *C64* emuliert. Aber die Art und Weise wie er es tut, ist revolutionär und zukunftsweisend. *JaC64* ist komplett in *Java*²⁵ geschrieben und läuft deshalb auf jedem System, auf dem eine *JVM*²⁶ installiert ist. Damit stellt der Emulator keine Voraussetzungen an die Plattform oder das System. Es wird lediglich die virtuelle Maschine vorausgesetzt. Der Emulator muss nicht einmal installiert oder heruntergeladen werden, er lässt sich bereits zusammen mit einem Spiel aus einem Webbrowser heraus starten.²⁷

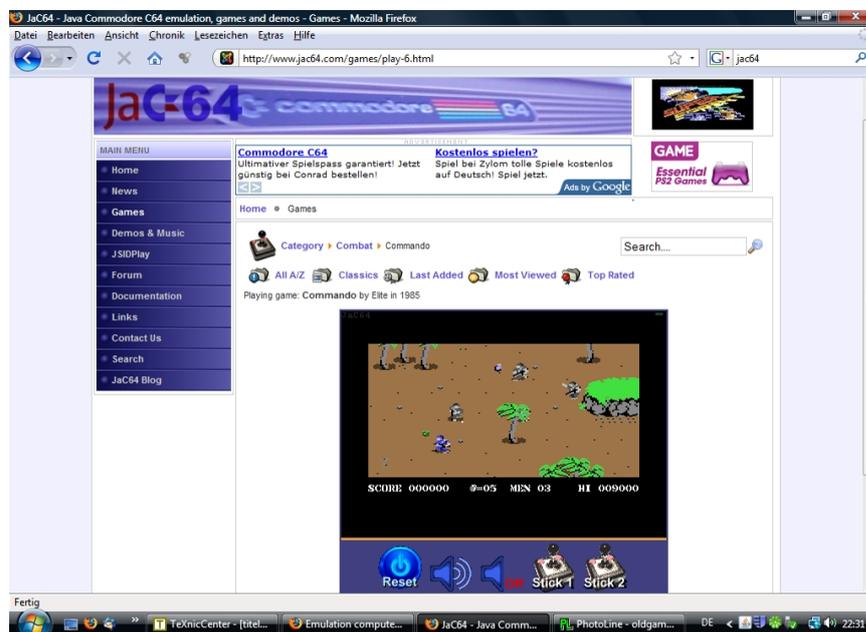


Abbildung 2: Das Spiel „Commando“ durch JaC64 in einem Webbrowser emuliert. Quelle: *eigener Screenshot*

3.5 JPC

JPC ist ein Projekt²⁸ der *Oxford University* und stellt ähnlich wie der *JaC64* eine zukunftsweisende Entwicklung dar. *JPC* ist ein x86-System-

²⁵ *Java* ist eine Programmiersprache die nicht in eine spezielle Maschinsprache übersetzt wird, sondern in sogenannten Bytecode für eine virtuelle Maschine. Weitere Informationen findet man unter: <http://www.java.com/>

²⁶ **Java Virtual Machine**. Das ist die virtuelle Maschine, für die *Java*-Programme geschrieben werden.

²⁷ Die Webseite und weitere Informationen findet man unter: <http://www.jac64.com/home/index.php>

²⁸ Projekthomepage: <http://www-jpc.physics.ox.ac.uk/index.html>

Emulator, der auf *Java* basiert. Damit wird die fast grenzenlose Plattform-unabhängigkeit garantiert und es werden völlig neue Wege eröffnet. Auf dem mit *JPC* emulierten x86-System lassen sich alle x86-basierten Betriebssysteme ausführen und entsprechend auch alle zugehörigen Programme starten. Das erleichtert das Erhalten von Computerspielen in vielerlei Hinsicht. Möchte man nun ein beliebiges Spiel spielen, so sucht man sich den besten bekannten Emulator dafür aus. Danach führt man mit *JPC* das Betriebssystem aus, das der Emulator erfordert, startet darauf den Emulator und schon kann das Spiel darauf ausgeführt werden. Damit werden bekannte Programme, wie die virtuelle Maschine „*Virtual PC for Windows*“²⁹ mit der man mehrere Betriebssysteme auf einem Rechner ausführen kann und die Anwendung „*WINE*“³⁰, mit der man unter Linux Windows-Anwendungen ausführen kann, nicht länger benötigt. Es sei ausdrücklich darauf hingewiesen, dass *JPC* keinen PC zur Ausführung benötigt, sondern lediglich ein System, das die *JVM* unterstützt, beispielsweise ein mobiles Telefon. [Newman and Dennis, 2009]

4 Eignung von Emulatoren zur Langzeitarchivierung

4.1 Erörterung

Die vorgestellten Emulatoren zeigen, dass es bereits jetzt möglich ist, mehrere Epochen der Spiele-Geschichte zu erhalten und wieder zum Leben zu erwecken. Es ist auch das Bestreben erkennbar, Emulatoren so zu entwickeln, dass sie einfacher zu portieren sind und damit selbst länger zu erhalten sind. Doch sollte man sich an dieser Stelle fragen, ob Emulatoren an sich bereits das beste Mittel für die Langzeitarchivierung von Computerspielen sind. Ein großer Nachteil von Emulatoren ist, dass sie häufig eine sehr große Rechenlast erzeugen. Sie können niemals die Rechenleistung des emulierenden Systems erreichen. Das heißt, dass wenn das emulierende System eine Rechenleistung von 1000 MHz erreicht, der Emulator keine CPU mit annähernd 1000 oder mehr MHz emulieren kann. Allerdings kann man durch trickreiche Programmierung immerhin einen mehr oder weniger guten Wirkungsgrad erzielen. Zurzeit sieht es so aus, dass einige sogar sehr alte Spiele auf heutigen High-End-Rechnern nicht flüssig emuliert werden können³¹. Andererseits gibt es Spiele, die viel zu schnell emuliert werden. Das hängt stark von der Architektur des emulierten Systems ab. Die Emulation eines Spieleautomaten mit mehreren Prozessoren ist aufwändiger als beispielsweise

²⁹weitere Informationen unter: <http://www.microsoft.com/windows/virtual-pc/>

³⁰weitere Informationen unter: <http://www.winehq.org/>

³¹Quelle: <http://mamedev.org/devwiki/index.php/FAQ:About>

die Emulation eines einfachen Pentium II Rechners. Dann muss der Emulator die Funktion bereitstellen, Frames pro Sekunde zu limitieren bzw. die emulierte CPU herunterzutakten. Weiterhin muss beim Emulatorenbau darauf geachtet werden, dass keine Lizenzen verletzt werden, denn nicht jedes Computersystem wird automatisch nach einer gewissen Zeit vom Hersteller freigegeben. Unter Umständen entstehen an dieser Stelle hohe Kosten für den Kauf von Lizenzen, die aber zunächst nur dem Emulator anzurechnen sind. Ein Emulator allein bringt keinen großen Nutzen. Die Spiele dafür müssen von den Herstellern ebenfalls zur Emulation freigegeben werden, da sonst leicht Urheberrechtsverletzungen entstehen. Die möglichen Lizenzierungskosten können eine große Hürde darstellen, da die bekannten Emulatoren bisher in der Regel aus Fanprojekten hervorgegangen sind und keine offizielle finanzielle Unterstützung erhalten haben. Sollten plötzlich Lizenzierungen notwendig werden, so könnte man die Machbarkeit des Konzeptes der Emulation, wie es bisher praktiziert wurde, leicht in Frage stellen. Da aber offensichtlich bisher sehr gute und mächtige Emulatoren aus Eigeninitiative entstanden sind, halte ich es für machbar, dass diese Entwicklung mindestens so fortgeführt wird, wenn nicht sogar bald finanzielle Mittel in diese Richtung fließen, weil das Thema der Langzeitarchivierung immer akuter wird. Beispielhaft ist das Projekt *KEEP*³², das mit mehreren Millionen Euro unterstützt wird und zum Ziel hat, eine universelle Datenplattform zu kreieren, auf der das digitale kulturelle Erbe der Menschheit erhalten werden kann. Seit einigen Jahren entstehen auch Computermuseen, die sich um die Erhaltung von Soft- und Hardware bemühen³³. [nestor, 2009]

Das vorteilhafte an Emulatoren ist, dass man einmal mit großem Aufwand und fundierten Spezialkenntnissen eine Anwendung erstellen muss, man danach aber fast keinen Arbeitsaufwand mehr hat. Vor allem können mit einem Emulator gleich mehrere Spiele emuliert und damit erhalten werden, was den Folgeaufwand (beschränkt sich auf das zur Verfügung stellen der Spiele) konstant klein hält, während beispielsweise bei der Migration jedes einzelne Spiel aufs Neue migriert werden muss. Somit ist es auch nicht notwendig, viele verschiedene Migrationsversionen zu speichern, sodass der Emulationsansatz generell sparsamer mit Speicherplatz umgeht als andere Erhaltungsmaßnahmen. Ein weiterer sehr großer Vorteil der Emulation ist, dass alle zu erhaltenden Daten im Originalzustand belassen werden und damit ein hoher Grad an Authentizität garantiert wird. Auf diese Weise wird sichergestellt, dass man sich künftig zu jeder Zeit stets auf das Originaldo-

³²**Keeping Emulation Environments Portable**, ein europäisches Projekt zur Erhaltung digitaler Artefakte. Weitere Informationen findet man unter: http://cordis.europa.eu/fetch?CALLER=FP7_PROJ_EN&ACTION=D&DOC=1&CAT=PROJ&QUERY=011f37a73b31:61ba:091d22f8&RCN=89496

³³siehe beispielsweise: <http://www.digitalgamearchive.org/home.php>,
<http://www.computerspielemuseum.de/> oder
<http://www.computermuseum-muenchen.de/>

kument beziehen kann und nicht befürchten muss, dass Spiele beispielsweise durch „grafisches Aufpolieren“ während einer Migration verfälscht wurden [Borghoff et al., 2006]. Authentizität ist für historische Belege natürlich sehr wichtig. Eine Studie³⁴ hat jedoch ergeben, dass von Benutzern das emulierte Spiel nicht zwangsläufig dem migrierten Spiel vorgezogen wird. Man muss also stets den Zweck der Erhaltung im Auge behalten, um das richtige Mittel dazu zu wählen.

4.2 Ausblick und Portierbarkeit

Es wurde bisher gesagt, dass sich das aufwändige Entwickeln eines Emulators dadurch auszahlt, dass sich der Folgeaufwand sehr gering hält. Doch was tut man, wenn ein Emulator sehr erfolgreich entwickelt wurde, nach einer gewissen Zeit aber die Grundlage des Emulators - das emulierende System - nicht mehr verfügbar ist? Im Grunde gibt es zurzeit drei Ansätze dafür, die ausführlich in [Borghoff et al., 2006] beschrieben werden.

Der erste eher triviale Ansatz ist der, für den Emulator wieder einen Emulator zu schreiben und so fortlaufend eine Kette von ineinander geschachtelten Emulatoren zu erzeugen. Wie bereits angesprochen ist aber ein einziger Emulator bereits sehr rechenlastig, sodass der Ansatz einer Emulatorkette aus Effizienzgründen nicht durchgeführt werden sollte.



Abbildung 3: Emulatorkette: Jedes Programm P hat benötigt einen Emulator E, der auf der Maschine M ausgeführt werden kann. Die Maschine M könnte wiederum durch einen weiteren Emulator emuliert werden usw.
Quelle: *eigene Grafik*

Ein weiter Ansatz sieht vor, alle Emulatoren in einer gemeinsamen Sprache zu beschreiben und umzusetzen, sodass man später nur noch für alle Folgesysteme jeweils ein Programm für die gemeinsame Spezifikation entwickeln muss. Dieser Ansatz ist weit besser als der erste, nur hat man sich

³⁴nachzulesen unter: <http://worldcat.org/arcviewer/1/OCC/2007/08/08/0000070513/viewer/file1628.html#feature1>

bisher auf keine gemeinsame, übergreifende Spezifikationsvorschrift geeinigt, um das Konzept so umzusetzen.

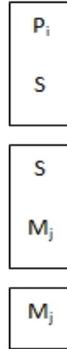


Abbildung 4: Jedes Programm P wird in einer Spezifikationssprache abgefasst. Für viele Maschinen M gibt es jeweils ein Programm S , das die Spezifikation von P für M übersetzt. Quelle: *eigene Grafik*

Der dritte Ansatz ähnelt dem zweiten und sieht vor, alle Emulatoren in der gleichen Programmiersprache zu implementieren und diese auf einer virtuellen Maschine auszuführen. Dies ist der Ansatz von *Java* und der *Java Virtual Machine*. Der Vorteil liegt hierbei darin, dass man auf Zukunftssystemen nur die virtuelle Maschine ausführen muss und dann sämtliche Programme, die jemals für diese Maschine geschrieben wurden, vorausgesetzt, dass das Ansprechen spezieller Peripherie portabel verallgemeinert wurde, ausführen kann.



Abbildung 5: Jedes Programm P wird für eine virtuelle Maschine VM geschrieben. VM kann auf vielen Maschinen M ausgeführt werden. Quelle: *eigene Grafik*

Dieses Konzept hat sich bereits bei ganz normalen Anwendungen sehr bewährt, da man von Betriebssystemen weitgehend unabhängig ist. Nach diesem Konzept gibt es nun auch die ersten Emulatoren wie dem *JaC64* wie auch *JPC* und es zeichnet sich der Trend ab, dass man den Ansatz der

virtuellen Maschine forciert verfolgt werden wird.

Im Sinne der Portierbarkeit stellt der Ansatz der virtuellen Maschine momentan die zukunftssicherste Lösung dar. Emulatoren wie *MAME* und *MESS* konnten zwar aufgrund einer vereinheitlichten Sprachbibliothek leicht auf andere Betriebssysteme portiert werden, allerdings war das eher ein brauchbarer Nebeneffekt, als Absicht, wie die Entwickler berichten³⁵. Ein Programm für eine virtuelle Maschine ist hingegen völlig unabhängig von Betriebssystemen und Rechnerarchitektur, sodass man mit diesem Ansatz die flexibelste Portierbarkeit erhält.

³⁵nachzulesen unter: <http://mamedev.org/devwiki/index.php/FAQ:About>

5 Literaturverzeichnis

Literatur

- [Borghoff et al., 2006] Borghoff, U. M., Rödiger, P., Scheffczyk, J., and Schmitz, L. (2006). *Long Term Preservation of Digital Documents - Principles and Practices*. Springer Berlin Heidelberg.
- [Carless, 2004] Carless, S. (2004). *Gaming Hacks*. O'Reilly. Allgemeines zu Emulatoren und Portabilität.
- [Hoffmann, 2008] Hoffmann, D. W. (2008). *Software-Qualität*. Springer Berlin Heidelberg. Kapitel 1.1: Aufbruch in das ubiquitäre Computerzeitalter.
- [Huth and Lange, 2004] Huth, K. and Lange, A. (2004). Die Entwicklung neuer Strategien zur Bewahrung und Archivierung von digitalen Artefakten für das Computerspiele-Museum Berlin und das Digital Game Archive. Technical report, Digital Game Archive e.V. Zu finden unter: http://www.archimuse.com/publishing/ichim04/2758_HuthLange.pdf.
- [Kohler, 2005a] Kohler, C. (2005a). *Retro Gaming Hacks*. O'Reilly. Kapitel 7: Playing with DOS.
- [Kohler, 2005b] Kohler, C. (2005b). *Retro Gaming Hacks*. O'Reilly. Kapitel 3: Playing Arcade Games on Your Computer.
- [Kohler, 2005c] Kohler, C. (2005c). *Retro Gaming Hacks*. O'Reilly. Kapitel 4: Playing Classic Console Games.
- [nestor, 2009] nestor (2009). *nestor Handbuch - Eine kleine Enzyklopädie der digitalen Langzeitarchivierung*. Verlag Werner Hülsbusch. Kapitel 8.4 von Stefan E. Funk.
- [Newman and Dennis, 2009] Newman, R. and Dennis, C. (2009). *JPC: An x86 PC Emulator in Pure Java*. O'Reilly. Kapitel 9 aus „Beautiful Architecture - Leading Thinkers Reveal the Hidden Beauty in Software Design“.
- [Picot et al., 2008] Picot, A., Zahedani, S., and Ziemer, A. (2008). *Spielend die Zukunft gewinnen*. Springer Berlin Heidelberg. Beiträge vieler verschiedener Autoren zum Thema „Computerspiele“, deren Entwicklung, Vermarktung, Einfluss uvm.
- [Rothenberg, 2000] Rothenberg, J. (2000). *Using emulation to preserve digital documents*. Koninklijke Bibliotheek. Kapitel 4: The concept of emulation, und ff.

- [Thomas and Stammermann, 2007] Thomas, W. and Stammermann, L. (2007). *In-Game Advertising - Werbung in Computerspielen*. Gabler. Kapitel 2: Wer nutzt Computerspiele - eine Zielgruppen-Analyse.
- [Westermann, 2008] Westermann, R. (2008). *Elektronische Spiele als Forschungstreiber*. Springer Berlin Heidelberg. Erschienen in „Spielend die Zukunft gewinnen“.
- [Wunderlich et al., 2009] Wunderlich, H.-J., Elm, M., and Kochte, M. (2009). *Zuverlässigkeit mechatronischer Systeme*. Springer Berlin Heidelberg. Kapitel 8: Bewertung und Verbesserung der Zuverlässigkeit von mikroelektronischen Komponenten in mechatronischen Systemen, S. 412ff.