

Códigos & Circuitos *its*: obstáculos, desvios e atalhos no ensino e pesquisa (através) de *softwares* e *hardwares* Vintage¹

Stefan Höltgen²

Tradução: Carolina Berger

Resumo: O autor trabalha como pesquisador e professor de História e Arqueologia da computação. Uma vez que atua com antigas tecnologias de hardware e software, questões operacionais relativas à preservação e ao *copyright* dos mesmos são um discurso constante e que deve ser considerado. Enquanto arquivistas, profissionais voltados à preservação e curadores de museus abordam esta questão como um obstáculo, os arqueólogos da tecnologia computacional estão focando em práticas atuais e históricas que circunscrevem estas leis, em tecnologias criativas e em práticas derivadas das leis de *copyright* (como práticas da cultura *hacker*, distribuição da informação, descompilação de aplicativos, entre outras). No presente artigo, o autor discute alguns pontos de vista de abordagem arqueológica sobre obstáculos relacionados à preservação de softwares e hardwares. Como curador de um laboratório de pesquisa com computadores *vintage* e seus softwares, ele enfrenta problemas e oportunidades resultantes das leis de *copyright* surgidas na cena da “computação retro”. Por outro lado, surgem questões relacionadas a metodologias técnicas e práticas, ambas com consequências epistemológicas relevantes à discussão do assunto.

Palavras-chave: Arqueologia computacional. Arquivo. Copyright. Cultura hacker.

Abstract: The author is working as a researcher and teacher for computer history and computer archaeology. Since the latter deals with operative vintage computer hardware and software the issues of software preservation and copyright problems are a constant discourse to be considered. Where archivists, preservers and museum curators see an obstacle in this, computer archaeologists are focusing on the historical and actual practices of circumvent these laws and on the creative technologies and practices that derive from the copyright laws (like hacker practices, information distribution, decompiling software). In the paper the author discusses some of these computer archaeological sights on the obstacles of hardware and software preservation. As the curator of a lab with vintage computers and their software he faced the problems and opportunities which copyright laws bring to retro computing scenes – and about the technical methods and practices as well as the epistemological consequences concerning with this topic.

Keywords: Computacional archaeology. Archive. Copyright. Hacker culture.

¹ Este artigo baseia-se em minha fala na Conferência *Save Game – Legal Challenges in Game Preservation*, em 22 de abril de 2015 (Berlim): <<http://txt3.de/qws2015>>. (20.06.2016).

² Dr. Stefan Höltgen, do Instituto de Musicologia e Estudos Midiáticos (*Institute of Musicology and Media Studies*), da Universidade Humboldt, Berlim. E-mail: stefan@hoeltgen.org | Site: <http://www.stefan-hoeltgen.de>.

Atualmente, eu estou pesquisando a arqueologia dos primeiros microcomputadores e seus sistemas de programação³ no *Departamento de Musicologia e Estudos Midiáticos* da Universidade Humboldt em Berlim. Entre meus tópicos de pesquisa e ensino estão os usos históricos dos microcomputadores de 4, 8 e 16 bits e outros hardwares eletrônicos, analógicos e digitais. Objetivo elaborar uma epistemologia do uso privado dos computadores, no passado e no presente. Ao longo de minha pesquisa, eu enfrentei inúmeros problemas, questões e discursos sobre preservação de software e *hardware* e suas complicações legais. No presente artigo, eu descrevo três principais tópicos de minha pesquisa.

Portanto, minha pesquisa demanda não somente um olhar sobre abordagens acerca de computadores, mas também códigos, circuitos e diagramas, e sobre *design* de softwares (até *design* de chips). A leitura destes aspectos dá-se a partir da proliferação e dos usos de computadores nas últimas décadas: do exame das camadas de tempo nos jogos computacionais, eu analiso o game *Space Invaders*, seus códigos e suas placas de circuitos em 2015⁴; para formular uma reavaliação da história canônica dos microcomputadores eu reconstruo o jogo computacional histórico *Tennis for Two* em 2012, com estudantes selecionados para a pesquisa⁵; e para escrever um artigo sobre a epistemologia midiática dos “portais” e da localização dos personagens dos jogos computacionais e/nos hardwares e softwares, eu abordo o antigo *arcade game: Ms. Pac Man*, em 2014⁶.

Códigos do passado



Figura 1. “E.T. – O extraterrestre”.

³ Vide: <<http://txt3.de/open-abstract>>.

⁴ HÖLTGEN, 2016, 51-69.

⁵ HÖLTGEN/MAIBAUM/RECH, 2012, 32-37.

⁶ Vide. HÖLTGEN, 2014, 104-134.

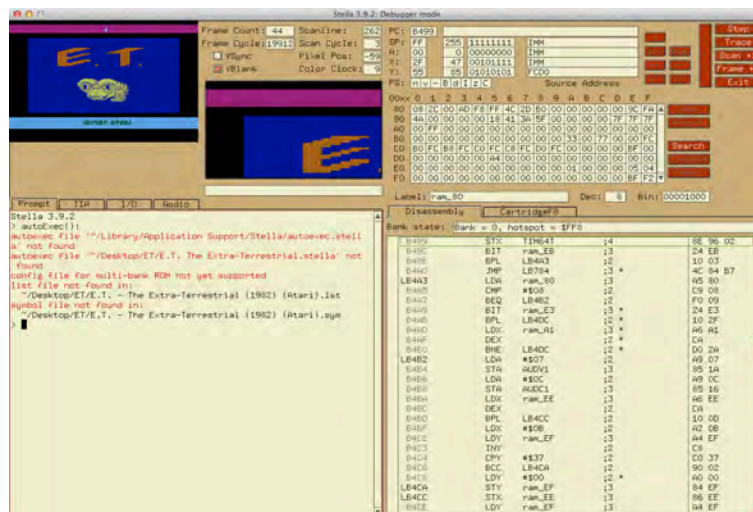


Figura 2. “Remontando E.T.”, com Stella Debugger.

Jogos de computador “retro” que copiam games antigos ou que são remakes para tecnologias contemporâneas são muito populares por dois motivos: por um lado, há uma tendência do público, atraído pelo retro; por outro, nos últimos anos, há um interesse renovado em seus códigos. Jogos como *E.T. O Extraterrestre* (para Atari VCS) foram a causa para uma escavação arqueológica em 2014⁷; *Príncipe da Pérsia* (para Apple II) teve uma cobertura midiática quando seu código fonte foi redescoberto no porão da casa do pai de um programador⁸ e foi publicado no *GitHub*⁹. Códigos de montagem desses jogos – acessados a partir de recursos ou decodificação binária – estão complementando uma história alternativa dos jogos de computador e desafiando as conhecidas históricas econômicas dos anos 1970 e 1980. Estas descobertas “revelam” a história das práticas hackers, dos algoritmos e das conexões históricas que atravessam a história da computação.

Space Invaders, da Taito, de 1978, que tornou-se o primeiro *arcade game* (fliperama), baseado no microprocessador e na montagem de códigos fonte, é um perfeito exemplo para esta história alternativa: combina antigas e novas tecnologias analógicas de TTL (*Transistor-Transistor Logic*) baseadas em circuitos integrados para propósitos gráficos e sonoros, com microprocessadores de tecnologia moderna (Intel 8080). *Space Invaders* foi codificado a partir de um conjunto de linguagens de código fonte que levou a inúmeras questões relacionadas com baixa memória e limitada velocidade de

⁷ Vide. HÖLTGEN, 2015, 56-63.

⁸ Vide. MECHNER 2012a.

⁹ Vide. MECHNER 2012b.

processamento, aspectos estes que se tornaram significantes para minha pesquisa acerca da necessidades de processamento e os efeitos entre códigos sonoros e circuitos analógicos destes sons¹⁰.



Figura 3. *Space Invaders*.

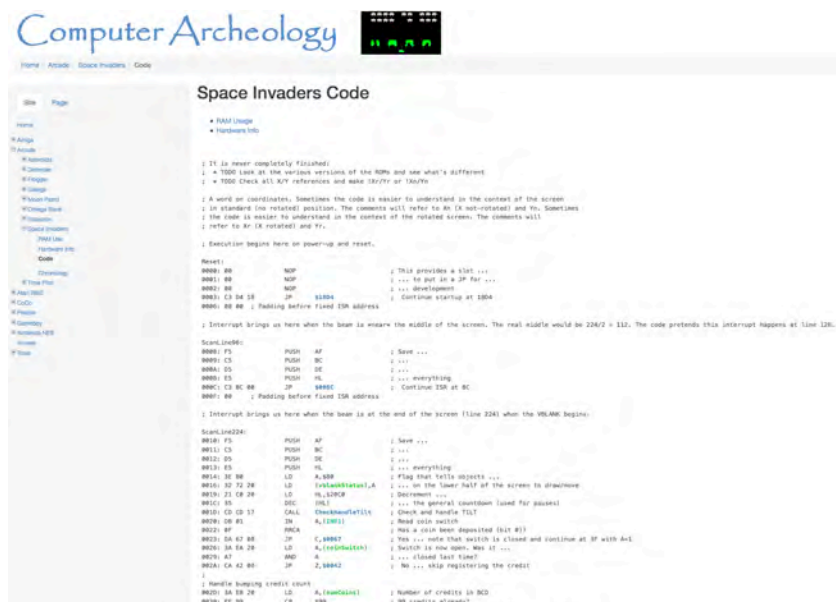


Figura 4. Arcade *Space Invaders*: remontagem em código de operações Z80.

¹⁰ Vide. HÖLTGEN 2016.

Para examinar o tempo de operações de um jogo, eu tive que analisar seus códigos. O *copyright* do código fonte de *Space Invaders* pertence à Taito, e nunca foi publicado devido à atual popularidade e sucesso comercial do jogo. Apesar de existir na internet uma coleção de códigos fonte com códigos *Space Invaders*: em *computerarchaeology.com* os “hobbystas” amadores estão desmontando jogos populares do passado e publicando seus códigos como arquivos textuais. Para evitar problemas legais relacionados aos *copyrights* do jogo *Space Invaders*, eles usam uma estratégia: os códigos de operação (comandos do processador) do microprocessador Intel 8080 é reduzido aos códigos de operação compatíveis com o processador Z80Processor, já que o Z80 foi o sucessor e competidor direto do processador Intel 8080. Mesmo que suas instruções de programação pareçam diferentes, o conjunto de códigos de operação e funções por trás deles são as mesmas. Então, os hackers publicam os códigos de *Space Invaders* com instruções de programação (mnemônicas) do Z80¹¹. Os programadores, então, reconheceriam esta estratégia e seriam capazes de construir conclusões para a plataforma Intel 8080 e ainda podem traduzir o código de Z80 para um código do Intel 8080 para um programa de execução ao usar os mesmos em máquinas e imoladoras do próprio 8080.

Apesar disto, existem obstáculos, não legais, mas sobretudo técnicos que os quais são possíveis de ler nos comentários sobre estes códigos. Para prevenir as “desmontagens”, os códigos foram encobertos através de diferentes técnicas de programação pelos seus programadores originais. Este método leva a uma decodificação dos códigos que contém partes distorcidas, sem sentido, que podem não funcionar se forem novamente reunidas ou codificadas. Estas distorções são uma importante parte da arqueologia da análise computacional, pois, podem demonstrar como a técnica cria um sistema de proteção de *copyright* através de um constante diálogo entre *hackers* e detentores de *copyright*. Mas, quando o código em si, e não o discurso sobre o código, é o centro das atenções das pesquisas de arqueologia computacional, alguns obstáculos intransponíveis podem passar a existir.

¹¹ Vide. HURST/SEVALLIUS.

Circuitos de jogos analógicos



Figura 5. Instalação *Tennis for Two* (Jogos para Dois), em 1958.

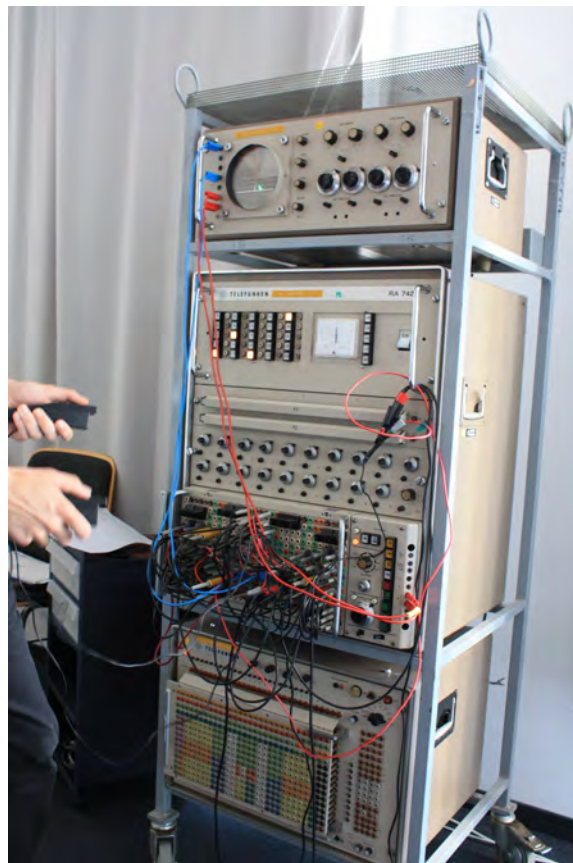


Figura 6. Recriação de "Tennis for Three", 2012.

Pesquisas em arqueologia computacional focam em vestígios e traços tecnológicos ocultos. Um exemplo é a tecnologia de computação analógica, cujos traços foram inteiramente perdidos no final dos anos 70, com a ascensão dos microprocessadores. Como demonstrado em *Space Invaders*, nem todos os jogos eletrônicos são puramente digitais (ou mesmo os videogames, como são comumente chamados¹²). Entre os jogos eletrônicos estão os jogos computacionais (*computer games*) para computadores digitais que são salvos como códigos alfanuméricos, em mídias de armazenamento de dados e outros tipos que são implementados em chips de memória ROM e módulos ROM para computadores e plataformas voltadas especificamente para games.

A terceira categoria – os videogames – são plataformas eletrônicas híbridas para a geração e manipulação de raios catódicos em CRTs (como o Magnavox's *Odyssey*) e uma quarta são os jogos para computadores analógicos. *Tennis for Two*, construído em 1958 por William Higinbotham no *Brookhaven Laboratories* é um dos exemplos¹³. Este (ou mais precisamente: o discurso sobre este game), tornou-se popular nos anos 80, não entre os jogadores, mas principalmente entre historiadores de games, por ser reconhecido como o primeiro jogo computacional.

Para além da certeza (ou da relevância) em relação à sua atribuição como o “primeiro” na pesquisa arqueológica, *Tennis for Two* tornou-se significativo em pesquisas de arqueologia computacional, pois, o discurso sobre a questão “ser o primeiro” provocou alguns interessantes reflexos históricos. O último deles foi um projeto de reconstituição: o *Museum of Electronic Games & Art (M.E.G.A.)* reconstruiu o jogo com uma distinta placa analógica, em 2011¹⁴. Esta não foi a mesma implementação analógica, ao passo que foi programada em uma placa com partes distintas dedicadas a fornecer apenas o game *Tennis for Two*¹⁵ – e assim resgatou o significado dos computadores analógicos programados livremente: prover uma configuração para o *design* de circuitos analógicos.

¹² Para uma diferenciação e definição dos termos “jogos computacionais” e “vídeo game” ver: PIAS, 2001, 105-111.

¹³ Para um panorama sobre jogos computacionais antigos ver: ULMANN, 2014.

¹⁴ MEGA 2011.

¹⁵ Higinbotham usou sua implementação do modelo computacional analógico de Donner 30, de 1958, com um osciloscópio.



Figura 7. Magnavox Ad para *Odyssey*, com *Table Tennis*.



Figura 8. Fliperama de *Atari Pong*.

Tennis for Two nunca seria mencionado na história dos jogos computacionais se não fosse um testemunho dos conjuntos das leis de copyright. Em 1983, Ralph Baer, o inventor do videogame *Odyssey* teve que defender seu jogo *Table Tennis* contra a Nintendo, que queria apagar sua patente para construir o *Table Tennis*, usando como exemplo jogos tais quais *Baer* e *Magnavox* que não pagavam taxas¹⁶. Uma reportagem

¹⁶ Vide LANGSAW 2014.

na revista *Creative Computing* sobre *Tennis for Two* foi usada como evidência¹⁷. Versões mais antigas de *Baer* e de *Magnavox* processaram a *Atari* pela sua versão de jogo *Pong*, por razões similares. *Tennis for Two* parece ter alcançado uma versão bastante popular nos anos 70, e no início dos anos 80. Mas o ponto interessante aqui é a diferença surgida entre os jogos *Surface* e os de *Subface* a partir destes exemplos: mesmo que a versão *Surface* dos três jogos (*Tennis for Two*, *Table Tennis*, e *Pong*) fossem similares, as versões *Subface* eram drasticamente diferentes. Desta forma, *Tennis for Two* tornou-se um importante artefato histórico para revelar sua indústria e passado.

Este discurso e suas implicações tecnológicas foram um dos motivos pelos quais eu e outros estudantes optamos por reimplementar o jogo *Higinbotham*, em 2012, no computador analógico *Telefunken RA-742*. Nossa questão principal era: seriam estes jogos (especialmente *Tennis for Two*) "invenções" da indústria eletrônica ou computacional? Seria relevante a questão judicial sobre quem o inventou primeiramente? Ou talvez fosse simplesmente resultado de problemas mais gerais, para serem resolvidos em relação aos computadores? Especialmente quando se trata de problemas balísticos (como *Tennis for Two*), há simulações de computador muito precoces das forças armadas dos EUA.

Utilizamos circuitos típicos das equações balísticas como as que podem ser encontradas em manuais de computadores analógicos como *RA-742*¹⁸. Nós os ativamos (ou interativamos) com dois gatilhos e chegamos ao nosso clone de *Tennis for Two*. O jogo foi publicamente lançado em 2013 (no *Computerspielmuseum Berlin*). A discussão da exposição foi: seria os jogos de tênis um circuito específico, um software ou somente uma materialidade implementada que não poderia ser protegida por patente? A ação judicial de 1983 não solucionou tal problema.

Materialidade computacional “retro”



Figura 9. CPU MOS 6502, de 1975 (CC BY-SA 3.0, <<http://txt3.de/6502-ic>>.).

¹⁷ Vide ANDERSON 1983.

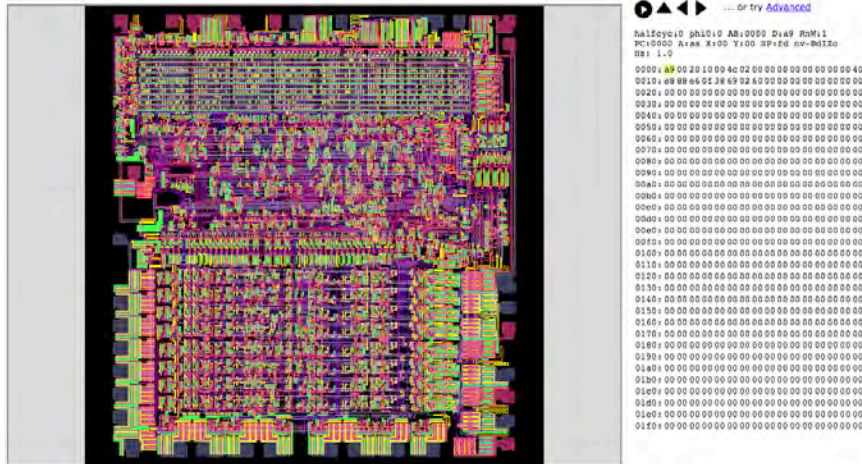
¹⁸ Vide TELEFUNKEN.

The Visual 6502

[FAQ](#) [Blogs](#) [Links](#)

This simulator uses HTML5 features only found on the latest versions of browsers and needs lots of RAM. If you have trouble, please [check compatibility](#).

Keyboard controls: 'z' to zoom in, 'x' to zoom out, 'n' to stop the simulation.
 Mouse controls: Left-click and drag to scroll around (when you're zoomed in.)
 More information in the [User Guide](#).



Source code is available on [github:visual6502](#). Use the online [emu_lator_and_assembler](#) from the easy6502 tutorial and [disassembler](#) from e-tradition.net. For in-depth 6502 investigation and some more advanced features, try our [Advanced](#) page.

Figura 10. “Visual 6502” Javascript Simulation.

A decadência dos históricos hardwares computacionais é uma questão básica e um tópico do discurso museológico sobre preservação de software, pois, o hardware é necessário para a aplicação do software. O conjunto operacional ou as partes dos circuitos (especialmente circuitos de larga escala integrados [LSI]), depois de anos e décadas estão tornando-se disfuncionais em relação a suas publicações. Quando as partes sobressalientes se tornam raras, as formas de solução também são raras. Soluções como FPGAs (*Field Programmable Gate Array*) ou replicações TTL e emulações de *software* são uma possibilidade de uso para os aqui chamados *vintage softwares*¹⁹. Mas, os *designs* dos circuitos programados precisam ser conhecidos para serem construídos e programados. Projetos como *visual6502.org* estão fornecendo fotografias dos programas inoperantes e analisando suas estruturas para sua futura imolação ou reconstrução em “*home brew fabs*” (produção cadeira)²⁰. A questão de *copyrights* não parece ter relação com os *hackers*, mas quando estas informações sobre os *chips* são utilizadas para soluções sobre preservação de hardwares profissionais, sempre há a necessidade de resolver primeiramente as questões legais.

¹⁹ Usar aplicativos emuladores para simular *hardwares* é outra saída. No entanto, este método também implica em diferentes problemas (Cf. LANGE 2016).

²⁰ Vide. <<http://txt3.de/homebrew-fab>>. (20.06.2016).



Figura 11. Amstrad's *CPC 6128*: Computador pessoal (1985). Fotografia: Bill Bertram.



Figura 12. "KC Compact" Amstrad CPC. Clone de GDR (*German Democratic Republic*) (1989). Fotografia: Enrico Grämer.

Os direitos autorais de *design* de circuito por décadas têm sido ignorados pelos hackers e engenheiros, especialmente dos países do antigo bloco oriental que copiaram os chips ocidentais LSI e os microprocessadores. Quando não mais puderam analisar as estruturas defasadas, eles reconstruíram cópias funcionais TTL (um exemplo é o chip *Gate Array* da empresa *Amstrad CPC* e seu clone oriental *KC Kompact*). A estrutura do chip *Gate Array* CPC ficou desconhecida até este ano²¹; copiar seu comportamento a partir da lógica de software foi a primeira alternativa para a criação dos emuladores

²¹ Hackers do grupo visual6502.org conseguiram romper as tecnologias de Gate Array no início de 2016 (<<http://txt3.de/ga-decapped>>. 20.06.2016). O detentor do copyright de Amstrad parou de reter informações sobre este chip (já que sua manufatura esgotou no final dos anos 80 e a companhia havia lucrado mais no início de sua produção).

CPC. A substituição da tecnologia de chips TTL, seria outro caminho: Chris Smith usou a mesma para o chip ULA dos computadores *Sinclair*²². Por sorte, Smith conseguiu a permissão dos detentores do copyright para analisar o *design* de circuito, publicou e reconstruiu o ULA com a tecnologia TTL²³ para prepará-lo para uma futura extinção do chip.

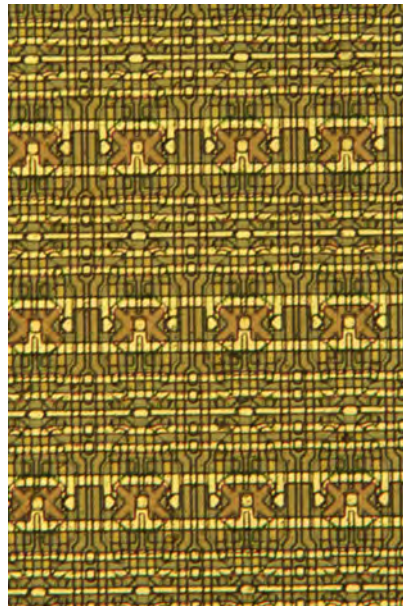


Figura 13. "2114" 4-quilobit SRAM: abertura química²⁴.

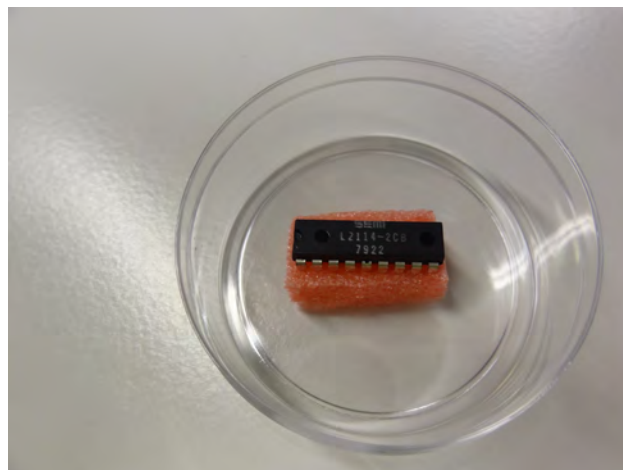


Figura 14. "2114" 4-quilobit SRAM: abertura.

Para contribuir com um livro sobre a epistemologia de "portais" como espaços de memória que direcionassem o fenômeno dos contínuos saltos das tecnologias nos

²² SMITH, 2010.

²³ SMITH, 2010, 247-251.

²⁴ Feito com a ajuda do grupo *Novel Materials*, do Departamento de Física da Universidade de Humboldt (Berlim).

jogos computacionais²⁵, eu utilizei a prática de análise do circuito físico, exatamente como os *hackers* do *visual6502.org* o fazem. Em cooperação com o grupo *Novel Materials* do *Departamento de Física* da Universidade Humboldt de Berlim, eu abri um chip “2114” DRAM com materiais químicos e fiz fotografias a laser de suas estruturas defasadas²⁶. Essa era a única forma de localizar os lugares reais e simbólicos dos personagens de jogos (como *Ms. Pac-Man* sobre o qual o capítulo por mim escrito, debruça-se). Este método era necessário porque as fotografias deterioradas dos chips DRAM só mostravam o famoso “1103” da Intel (por ser seu primeiro chip DRAM). Seus sucessores são largamente ignorados pelos historiadores de hardware, por serem considerados sem “originalidade”. Desde que o jogo de fliperama *Ms. Pac-Man* começou a utilizar o chip 2114, a máquina passou a ser produzida por muitos fabricantes, já que em 1979 alcançou o mercado, e seu sistema de segurança de dados não mostrava a estrutura de seu *hardware*. Uma aproximação de seus diagramas só seria possível se fosse buscada a abertura destes sistemas, realizada por mim. Por sorte, eu encontrei diferentes fontes na internet a partir das quais descobri instruções sobre como acessar estes sistemas.

As estruturas únicas deste chip são informações importantes – já que, não somente os dados do software (“o saber que”), mas também as estruturas de hardware (“o saber como”) são importantes para aprendermos sobre como um jogo de computador funciona quando é operacional. Ao passo que a maior parte da memória estrutural de um computador está oculta em caixas pretas, a abertura das mesmas e de seus conteúdos torna-se a práxis de uma descoberta autodeterminada. Vale mencionar que aprender sobre estas estruturas foi, de muitas formas, algo proibido no passado²⁷.

Conclusões

O problema da legalidade levanta duas questões em relação à pesquisa arqueológica computacional: primeiro, é um ponto de partida para questões arqueológicas específicas sobre a relevância dos diferentes discursos sobre a história da tecnologia; em um segundo nível, traz questionamentos sobre materiais com direi-

²⁵ HÖLTGEN, 2015, 107-134.

²⁶ O processo foi documentado. Informações no link: <<http://txt3.de/open-2114>>.

²⁷ Por exemplo, veja o caso de quando PlayStation 3: ele foi hackeado por George Hotz que tentou publicar informações sobre o seu sistema e a Sony ameaçou processá-lo (Vide. <<http://txt3.de/ps3-hotz>>. 20.06.2016).

tos autorais e usualmente força o uso destes materiais em contraste com as leis usuais relacionadas ao tema. Isto leva diretamente à recriação das práticas hackers que suspendem a história da tecnologia, concentrada somente em discursos. Portanto, a arqueologia computacional não somente demanda artefatos computacionais históricos operativos para examiná-la, mas sobretudo trata-se também das práxis materiais direcionadas ao uso das máquinas.

Infelizmente, a liberdade para ensinar e pesquisar não pode ser garantida desta forma devido às atuais circunstâncias legais, já que a ascensão da “computação retro” como uma prática lucrativa nos campos da computação e dos jogos, resulta uma revisão das leis de direitos autorais²⁸.

Enviado: 17 março 2017

Aprovado: 31 março 2017

²⁸ Vide. PITCHER 2015.

Referências

ANDERSON, John. Who really invented the video game?: there was Bell, there was Edison, there was Fermi, and then there was Higinbotham. In: *Creative computing, video & arcade games*, 1982. Vol. 1, No. 1.

HÖLTGEN, Stefan. It's more fun to compute!: retro-games als Wissensobjekte. In: Ann-Marie Letourneur; Michael Mosel; Tim Raupach. (Eds.). *Retro-games & retro-gaming*. Glückstadt: vwh, 2015, p. 49-66. Disponível em: <<http://migre.me/vM4dH>>.

_____. JUMPs durch exotische Zonen. Portale, hyperräume und teleportation in computern und computerspielen. In: Thomas Hensel; Britta Neitzel; Rolf F. Nohr. (Eds.). *The cake is a lie! Polyperspektivische betrachtungen des computerspiels am beispiel von ›portak*. Münster: LIT, 2015, p. 107-134. Disponível em: <<http://txt3.de/acad-jumps>>.

_____. Time Invaders. Zeit(ge)schichten in computerspielen. In: Höltgen, Stefan; VAN TREECK; Jan Claas. (Eds.). *Time to play: zeit und computerspiel*. Glückstadt: vwh 2016, p. 51-69.

HÖLTGEN, Stefan; MAIBAUM, Johannes; RECH, Matthias. Tennisspielen mit physik. In: *Retro*, No. 24, 2012, p. 32-37. Disponível em: <<http://txt3.de/t43, 20.06.2016>>.

HURST, Harry; SEVALLIUS, Patrick. Space Invaders. In: *Computer Archaeology*, 2016.

LANGE, Andreas. Playing in different times: herausforderungen und möglichkeiten der bewahrung von computerspielen. In: Höltgen, Stefan; VAN TREECK, Jan Claas. (Eds.). *Time to play: zeit und computerspiel*. Glückstadt: vwh 2016, p. 314-322.

LANGSHAW, Mark. Magnavox odyssey retrospective: how console gaming was born. In: *Digital Spy* (revista online), 2014. Disponível em: <<http://txt3.de/magnavox>>.

MECHNER, Jordan. Prince of Persia source code – found! In: *Jordanmechner.com*, 2012a. Disponível em: <<http://txt3.de/mech-pop>>.

MECHNER, Jordan. A running-jumping-swordfighting game i made on the apple ii from 1985-89. In: *GitHub*, 2012b. Disponível em: <<http://txt3.de/pop-source>>.

MEGA. T42 – *Tennis for two*. 2016. Disponível em: <<http://txt3.de/mega-t42>>.

PIAS, Claus. *Computer spiel welten*. Zürich/Berlin: Diaphanes, 2001.

PITCHER, Jenna. ESA Says preserving old games is illegal because it's 'hacking'. In: *IGN*, 2015. Disponível em: <<http://txt3.de/retro-hack>>. Acesso em: 20 jun. 2016.

SMITH, Chris. *The ZX Spectrum ULA: how to design a microcomputer*. Vale of Glamorgan: ZX Design and Media, 2010.

TELEFUNKEN. *Demonstrationsbeispiel Nr. 5. Ball im Kasten*. Disponível em: <<http://txt3.de/ball-in-box>>.

ULMANN, Bernd. *Signalspiele: mondlandung auf dem analogcomputer*. In: *Retro*, No. 32, 2014, p. 20-23.