Not Just One, but Three: An Experiment with Comparative Platform Studies

platform studies, home computers, video games, computer games

Markku Reunanen
markku.reunanen [a] aalto.fi
Senior university lecturer, Aalto University
Docent, University of Turku

Tero Heikkinen
tero.heikkinen [a] uniarts.fi
University researcher
The University of the Arts Helsinki

Adela Lujza Lučenič
lujza.lucenicova [a] aalto.fi
MA student, Aalto University

Elias Ruokanen
elias.ruokanen [a] aalto.fi
MA student, Aalto University

Heya Kwon
heya.kwon [a] aalto.fi
MA student, Aalto University

Yuxin Mu
yuxin.mu [a] aalto.fi
MA student, Aalto University

Xiangqian Huang
xiangqian.huang [a] aalto.fi
MA student, Aalto University



How to cite: Reunanen, Markku, Tero Heikkinen, Adela Lujza Lučenič, Elias Ruokanen, Heya Kwon, Yuxin Mu & Xiangqian Huang. 2026. ”Not Just One, but Three: An Experiment with Comparative Platform Studies”. *WiderScreen Ajankohtaista*. https://widerscreen.fi/numerot/ajankohtaista/not-just-one-but-three-an-experiment-with-comparative-platform-studies/

Platform studies, as established by the MIT Press book series, investigates how the hardware and software architecture of a computing platform shapes creative expression. This article extends the approach in a comparative direction by asking what advantages and challenges comparative platform studies might offer. While the tendency to examine platforms in isolation is understandable, such a narrow focus risks overlooking the relationships between them. Our primary research instrument is a detailed comparison of three games across three 8-bit platforms of the early 1980s – the Commodore 64, Sinclair ZX Spectrum and MSX – chosen for their different origins, distinct hardware designs and shared software titles. The case studies are complemented by a content analysis of existing hobbyist comparisons, including YouTube videos and websites, while the overall approach is informed by the long tradition of comparative literature. Rather than assuming the usefulness of a comparative approach, we assess its potential and limitations. We conclude by reflecting on the merits and challenges of comparative platform studies and proposing directions for future research.

Introduction

Platform studies is an emerging research tradition and a book series concerned with how computing systems shape cultural and creative expression. The book series was founded by Ian Bogost and Nick Montfort, and launched by MIT Press in 2009, although the term already existed before that. *Racing the Beam: The Atari Video Computer System* (Montfort & Bogost 2009) was the first book in the series, created “to promote the investigation of underlying computing systems and how they enable, constrain, shape and support the creative work that is done on them” (MIT Press n.d.). *Racing the Beam* set an influential early example for its successors, demonstrating a thorough close analysis of the Atari VCS and proposing a five-layer model of digital media – platform, form, interface, reception and community – as a vocabulary for that analysis. At its core, the approach insists that a computing platform’s hardware and software architecture is not merely a technical backdrop but an active determinant of what can be expressed.

This focus on computing architecture as a cultural determinant places platform studies in a distinctive position among neighboring fields. Unlike media archaeology, which focuses on genealogical and discursive histories of media forms across time, platform studies isolates a specific hardware and/or software environment and its relationship to culture. Unlike software studies, which focuses on software layers and often studies the general cultural implications of code and algorithms, platform studies covers both hardware and software environments while focusing more on a single platform or a family of them. Game studies, meanwhile, typically takes the game as its primary object and treats the platform as context rather than cause. That said, platform studies also considers environments beyond games and foregrounds the influence of the platform.

Since *Racing the Beam*, the series has expanded considerably in scope. While early volumes focused on gaming consoles like the Nintendo Wii (Jones & Thiruvathukal 2012) and the Nintendo Entertainment System (Altice 2015), the later titles broadened the frame to include other platforms including home computers and also software platforms. More recent volumes have pushed the boundaries even further, with James Malazita's *Enacting Platforms: Feminist Technoscience and the Unreal Engine* (2024) studying a game engine as a platform from the perspective of feminist, queer and race theories. While the book series, which currently comprises seventeen volumes, is the most visible institutional home of platform studies, the approach has also been taken up in journal articles, conference papers and edited collections. As the series grows, what qualifies as a platform and what qualifies as ”platform studies” to begin with, has become an increasingly open question.

There are critiques worth noting as we contribute our own perspective to platform studies. Dale Leorke (2012) argues that the series risks becoming a ”production line” applying a standardized formula, which is counterproductive to its wider goal. Malazita's *Enacting Platforms* (2024) raises deeper questions about the material and cultural limits of platforms themselves, and about what counts as a platform in the first place (cf. Bogost & Montfort 2009). Apperley and Parikka (2018) have also noted that reception and community tend to receive thinner treatment than the hardware and software layers in practice. Yet, one may ask the opposite: what remains of platform studies if it turns into common history of computing or cultural studies?

Another critical point concerns platform studies' tendency to lean toward culturally beloved and commercially successful machines. The series carries a certain Anglo-American and Japanese bias.

European platforms remain mostly absent from the series, despite their historical significance across large parts of the world. The same is true of Latin American computing history, where a rich culture of clone consoles and informal hardware ecosystems has gone largely unexamined (Penix-Tadsen 2024). Alison Gazzard's *Now the Chips Are Down: The BBC Micro* (2016) and *Minitel: Welcome to the Internet* (Mailland & Driscoll 2017) are currently the only exceptions to this rule.

Perhaps most fundamentally for our purposes, the approach has remained strictly focused on one platform at a time, treating each machine as a self-contained object of study and leaving the relationships between contemporaneous competing platforms largely unexplored. This study responds directly to that limitation by comparing three 8-bit platforms of the early 1980s – the Commodore 64, Sinclair ZX Spectrum and MSX – through close analysis of three games that ran across all of them. To distill the essence of the paper into a research question we ask: *what advantages and challenges does comparative platform studies entail?* The comparisons are coupled with content analysis of similar existing hobbyist endeavors, such as YouTube videos and websites. Our goal is to neutrally assess the potential of the approach, as opposed to merely confirming its expected usefulness, so any shortcomings will also receive considerable attention.

In the following section, we lay the foundation for the experiment by discussing existing comparative practices and introduce an initial model for our approach. Next follows discussion on the platforms chosen for the study, as well as details on the selection criteria. After the reader is familiar with the machines and their main technical features, we move on to the comparative study itself, where three games are compared across three platforms. Finally, we reflect on the merits and problems of comparative platform studies and provide future development ideas for implementing similar studies with different material than our own.

Toward Comparative Platform Studies

The history of computing encompasses a history of comparisons: to name just a few, computers, game consoles, programming languages, video games, text editors and operating systems have been pitted against each other in order to decide their pecking order. Companies have touted their solutions' superiority compared to their competitors, with faithful users tagging along and the press, in their turn, capitalizing on the phenomenon (Saarikoski & Reunanen 2014; see also Haddon 1988; Reunanen, Hoeberechts & Viljanen 2024). IBM vs. Digital, PC vs. Mac, Nintendo vs. Sony PlayStation – well-known examples are not hard to find, and at times the conflicts have even

been referred to as ”wars” (e.g. Tomczyk 1984). From this perspective, comparative platform studies can lightheartedly be considered as yet another manifestation of a long-standing cultural obsession.

Contemporary multiplatform computer press, such as *Computer and Video Games, Games Machine* and *ACE*, ostensibly offered an environment for platform comparison. Although in broad strokes these magazines provided a richer view over a number of computer platforms and their capabilities, the game reviews were often not very systematic or technically oriented in their comparisons. Computer hardware could be compared according to their spec sheet, highlighting countable features like the number of commands in their BASIC language dialects or even the octaves output by their respective sound chips (Reunanen, Hoeberechts & Viljanen 2024). Single-platform magazines like *Zzap! 64* or *Your Sinclair* had even less motive for comparison in their reviews, making a note if the game originated as a hit on a ”competing” format, sometimes treating such origins with suspicion.

Enthusiast Efforts

When the same game runs on a dozen different machines, the question arises: which version is the best? There can hardly be a single correct answer, but that has never stopped people from asking. Long before platform studies coalesced into an academic field, player communities were already conducting their own comparative analyses, informally but persistently, in magazine letter columns, on internet forums and YouTube videos. What they have built is a rich, if uneven, body of popular comparisons, one whose strengths and blind spots together give direction to where academic work should intervene.

The most visible format where comparisons take place today is the YouTube comparison video. Our content analysis covers 23 such videos from five channels: Nick930, Retro Core, Console Wars, Digital Foundry (DF Retro) and Gaming History Source. The channel Nick930 popularized a now-standardized *Let’s Compare* structure, i.e. one game shown running on multiple platforms in sequence, typically without voiceover. The footage is meant to speak for itself and the focus falls almost entirely on audiovisual fidelity. Variations of the format introduce different evaluative logics. Retro Core’s *Battle of the Ports* uses the arcade original as a fixed reference point against which home conversions are measured, while *Console Wars* stages comedic head-to-head rivalries between SNES and Genesis/Mega Drive ports, framing platform loyalty as cultural identity (cf.

Saarikoski & Reunanen 2014). Digital Foundry's *DF Retro* stands apart, adding voiceover narration, technical analysis and historical context that bring the format closer to scholarly rigor and, in doing so, reveal what the other channels lack.

Written comparisons form a secondary but significant body of material. The *Finnish Retro Game Comparison Blog* (FRGCB Dude, n.d.) offers detailed cross-platform reviews with structured scoring categories and screenshots, and collaborates with several YouTube comparison channels. Community forums, such as *Lemon64* for the Commodore 64, *World of Spectrum* for the ZX Spectrum, *AtariAge*, and *MSX.org* host organic comparison threads where users post screenshots, debate versions and flag emulation inaccuracies. These forums often function as correction layers for the video content.

What these popular comparisons share is a systematic bias toward what can be captured onscreen. Resolution, color palette, sprite detail, music and sound effects are examples of factors that tend to be reliably present in most cases. What is systematically absent is everything that resists screen capture: the tactile feel of original hardware and its controllers, the visual character of period CRT displays, code-level porting decisions, packaging, reception and context of each version's release. Heavy reliance on emulation compounds the problem, since emulators themselves introduce inaccuracies, such as wrong PAL/NTSC settings, incorrect audio timing and visual artifacts absent on original hardware, all of which are rarely acknowledged. *Gaming History Source* is a rare exception among comparison channels, explicitly disclosing that no individual can own all platforms and that emulation is, therefore, a practical necessity. Such transparency, however, remains uncommon. The gap is analytically revealing. It exposes the community's implicit criteria for what constitutes a ”good port”, namely audiovisual fidelity judged through screen-capturable evidence, while everything that shaped the actual experience of play falls outside the frame.

Yet, the audience does not passively consume these comparisons. Comment sections beneath popular videos contain nostalgic personal narratives tied to specific hardware, platform-loyalty arguments, technical corrections regarding emulation artifacts, and contributed knowledge about obscure versions. Viewers function as participatory co-comparators, extending and contesting the video's claims. A systematic analysis of such comments, coded along dimensions such as nostalgia, evaluation, correction and competition, could yield significant insight into how platform value is socially constructed. These discussions also propagate outward into forums, *Reddit* and *Discord*, generating secondary layers of comparative discourse.

The blind spots of popular comparison, namely absent technical depth, underdocumented emulation methodology and missing historical context, mark precisely the territory where academic work should intervene. But how should scholars approach cross-platform comparison rigorously?

Learning from Comparative Literature

To avoid reinventing the wheel, it is useful to learn from other established fields of study where a comparative approach already has a long tradition. Among various comparative studies, comparative literature is particularly relevant for us, as it examines relationships across distinct cultural contexts while providing conceptual tools for identifying both connections and differences between objects of study.

At a broader level, current comparative literature encourages researchers to examine global and interdisciplinary phenomena relationally instead of analyzing them in isolation. Applied to platform studies, this perspective aligns with the idea of understanding targeted platforms as part of larger networks of development, rather than treating individual platforms as individual systems. As Henry Remak (1961) argues, comparative literature explores literature beyond the boundaries of a single nation, language or discipline by examining its relationships with other forms of knowledge ”by extending the investigation of literature both geographically and generically”. Although computing platforms differ from traditional literary works, comparative literature offers relevant perspectives for platform analysis.

Among the various approaches associated with comparative literature, *influence studies* and *parallel studies* are particularly useful here. Influence studies focus on observable connections between targets, examining how ideas, forms or practices travel from one context to another. Guillén (1993, 47–50) states that a typical study begins with either the transmitter of the influence or the receiver who assimilates it. A third alternative is the interplay of influences and their contact. He also draws attention to the medium of influence, “the translator, the critic, the theatrical director, the magazine” that makes the “passage of text X to Y possible.” Parallel studies, in contrast, investigate targets’ similarities that emerge independently across different contexts, exploring why comparable structures or solutions appear despite the absence of direct contact. Guillén (1993, 69–71) outlines several different approaches, such as examining the socioeconomic

similarities between cultural contexts or the inherent possibilities and restrictions within the medium itself (e.g. the rhythmic and grammatical properties of poetry).

These approaches provide a useful foundation for thinking about relationships between platforms. In particular, influence studies can be adapted to examine historical and technical relationships between platforms. Researchers may investigate whether particular technical solutions, interface conventions or design strategies were adopted from earlier systems, whether through direct borrowing, competitive imitation or shared industrial practices. Influence studies can also be applied to software porting. In this context, a game released on one platform may be treated as a reference point against which later versions can be examined. Such comparisons make it possible to analyze how gameplay mechanics, audiovisual presentation, and technical implementations are preserved, transformed or reinterpreted when software moves between different environments.

In contrast, parallel studies offer a complementary perspective. Just like comparative literature studies, similar technology could emerge in separate social and cultural contexts. Direct influence is not always necessary to explain similarities between platforms. As a result, developers and designers sometimes arrived at similar solutions independently. Therefore, beyond tracing exclusively technical development, examining how gameplay mechanics, visual strategies, optimization techniques or design patterns emerge across different platforms is also crucial. Inspired by comparative literature, such comparisons help reveal broader relationships between technological constraints and creative expression, making parallel studies a particularly valuable tool for cross-platform analyses.

Complicated Comparisons

Montfort and Bogost (2009, 149) visualized the layered model of digital media in *Racing the Beam* where they placed the platform as the foundation everything else is built upon. Figure 1 is an extension of their work with comparisons between two sets of layers added. Even a quick look reveals how the comparative approach complicates the analysis and gives rise to new, non-evident questions: how did an existing platform affect the design of another? Does *code* refer to *source code* that may be compiled to different *machine code* for two platforms? What if the *interface* hides all the underlying layers and the users are, thus, provided the same experience in spite of platform and code differences?

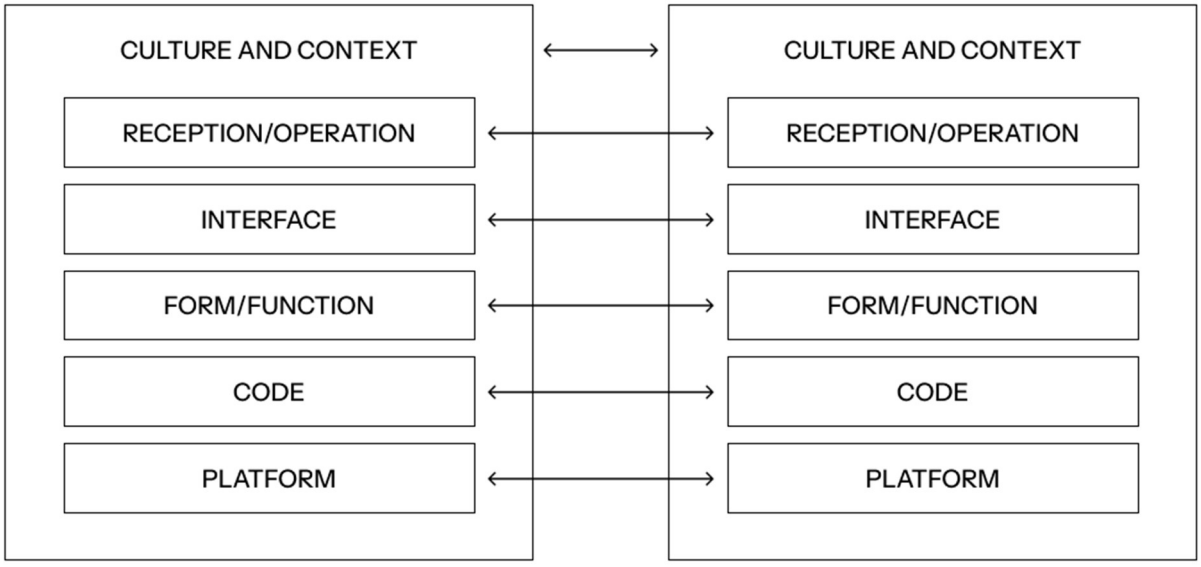


Figure 1. A comparative model of digital media, based on Montfort & Bogost (2009, 149).

The boundaries between the two boxes presented here are far from clear-cut, as development did not take place in isolated silos: hardware and software companies followed the trends of the time, adopting features and even directly cloning each other's solutions. The transition from text-based user interfaces to the WIMP metaphor is a well-known example from the interface level (e.g. Gere 2002, 124–38). The most complex part of the model, the backdrop of *culture and context*, completely dissolves the border if the comparisons do not take place between two different contexts to begin with – computing platforms have typically competed against each other in several markets at the same time.

As our focus here is at the platform level, we do not even aspire to address all the complications involved with the extended model. Even so, it is still crucial to avoid oversimplifications, such as considering the selected platforms as isolated sandboxes with no shared lineage or hiding the diverse complexity of the market. For instance, a computer that became dominant in one region may have been completely marginal in another. Forgetting such sidetracks happens too easily and leads to writing a history of the winners (cf. Therrien 2012; Reunanen, Joelsson & Saarikoski 2024).

Three Contenders

Selecting the computing platforms for this study was not a straightforward task, as the choice would inevitably affect the possible outcomes. After considering the alternatives, we decided on

three 8-bit machines from the early 1980s: the Commodore 64, Sinclair ZX Spectrum and MSX because of the following reasons:

1. They originate from three different locations (the US, UK and Japan).
2. Their hardware design clearly differs from each other.
3. Versions of the same software can be found for all of them.
4. They are well-documented, which facilitates in-depth analyses.

Modern computing platforms tend to be far more complex and also, by nature, increasingly closed ”black boxes”, which would only facilitate distanced or even superficial analysis. In terms of audiovisual capabilities, they follow the same paradigm consisting of hardware-accelerated polygon rendering, shaders and sample-based digital audio playback. Furthermore, today’s popular multiplatform game authoring tools, such as *Unity* or *Unreal Engine*, hide the details of the underlying platform even further – as such, tools themselves can also be considered as platforms (e.g. Salter & Murray, 2014; Malazita 2024).

What may, at first thought, appear as a clearly defined ”platform” is hardly ever such. In our case, too, there has been a great need to limit the discussion to a handful of configurations that form the core of the platform. The long-standing PAL/NTSC/SECAM rift divided even the same machine into different subspecies, which were not always compatible with each other. As games were tightly tied to the screen refresh rate (50 or 60 Hz), they would run at different speeds, often including the background music. The loading times between a tape, floppy drive and game cartridge differ greatly, which affects the user experience. In addition, later computer models with improved hardware capabilities further complicate the equation.

While the three selected platforms represent three geographic origins, the home computer industry of the time was already highly international. For instance, the ZX Spectrum is arguably British by design, yet its processor, the Zilog Z80, is American – to complicate the matter even further, various versions and clones of the chip were also produced outside the US. In the same vein, memory chips were sourced from different suppliers and even the computers themselves were manufactured in various countries regardless of the nominal origin of the company. Thus, calling the ZX Spectrum a British computer or the C-64 an American one is necessarily a simplification that hides many of the actual complexities of the supply chain.

Commodore 64: Are You Keeping up with the Commodore?

Commodore International's acquisition of the semiconductor chip manufacturer MOS Technology in 1976 gave it a pivotal edge in the development of an integrated hardware and software package (Bagnall 2005; Dillon 2015). Chips could be designed to meet the specifications of designers who had specific consumer needs in mind. The first machine born out of this collaboration was the relatively successful PET (Personal Electronic Transactor), released in 1977. It was followed three years later by the low-cost VIC-20, which the CEO Jack Tramiel marketed as being "for the masses, not the classes". Selling over a million units, it set the stage for its successor, the best-selling home computer model of all time: the Commodore 64.

Making its debut in August 1982 at the price of $595 in the United States, the Commodore 64 came packed with specialized in-house designed hardware that was exceptionally well-suited for gaming. While the clock speed of the MOS Technology 6510 processor was low compared to most competitors (0.985 MHz for PAL or 1.023 MHz for NTSC), the dedicated graphics and sound chip could perform many of the tasks that competitors' machines left to the processor.

The MOS Technology VIC-II (Video Interface Chip) graphics chip has a 16 kB address space shared with the processor (which can access the full 64 kB), 16 colors and a maximum resolution of 320×200 pixels in the monochrome hi-res mode or 160×200 in the multi-color mode, although these limitations can be worked around. The PAL version of the chip is tied to the 50 Hz screen refresh rate, while the NTSC one outputs a 60 Hz image. In practice, this difference meant that titles that were highly dependent on exact timing would not necessarily run correctly across American and European C-64s without modifications.

The special features of the VIC-II chip were heavily inspired by the arcade game boom of the late 1970s and early 1980s (Juul 2024, 20–2). Fast scrolling, sprites, collision detection and multiple colors had to be fitted into a desktop computer. To achieve the goal, the display is conceptually split into four layers: the border, sprites, characters and background. The background layer is a single color that serves as the foundation of the display, whereas the border is by default a single color frame that covers everything outside the active display area.

The sprite layer allows eight individual *sprites* (freely placeable graphical objects) to be displayed on the screen simultaneously. The default monochrome sprite size is 24×21 pixels and a colored one 12×21 pixels (half the horizontal resolution of a monochrome sprite). Eight sprite pointer

registers point to the blocks of memory where the sprite configurations are stored, while other registers define the sprite coordinates. Modifying these registers during the screen refresh allows for more sprites to be displayed, which is known as *multiplexing*, a technique that was commonly used in Commodore 64 games (Juul 2024, 146–8). Furthermore, the VIC-II provides pixel-perfect collision detection of sprites, which can be directly utilized in action games.

The character layer is built up of 8×8 (or 4×8 in low resolution modes) pixel sections, the contents of which were defined by the PETSCII character set in the text mode (see Reunanen, Heikkinen & Carlsson 2018). In other words, the character resolution of the whole screen is 40×25. While often used for static game elements or text, it was also repurposed to show simple sprites. The VIC-II also has a smooth scrolling feature which moves the screen by up to 7 pixels horizontally or vertically. To prevent screen tearing, software can be precisely synchronized with the screen's raster beam using techniques, such as raster interrupts and double buffering.

Another selling point of the C-64 was the highly versatile soundscape made possible by the MOS Technology 6581 SID (Sound Interface Device). Originally intended for a full-fledged synthesizer (Bagnall 2005), the SID is the first sound chip on a home computer to include an envelope generator (Dillon 2015). The chip has three voices, each with an oscillator with four waveforms (triangle, sawtooth, pulse and noise), an ADSR envelope generator and an amplitude modulator. The voices are fed to a multi-mode filter with low-pass, band-pass and high-pass outputs, with adjustable cutoff and resonance. The limitation of three voices gave rise to a distinct musical style characterized by the available waveforms, interleaving instruments on the same channel and fast arpeggiation that would often replace chords.

As to game controllers, the C-64 features two Atari-compatible 9-pin joystick ports, which facilitates two-player games. The hardware also supports analog paddles, even if games seldom made use of them, owing to their rarity and unsuitability for most genres. The feel of the built-in typewriter-style keyboard varies slightly across hardware revisions because of different spring mechanisms, but its layout remained uniform until the end of the machine's commercial lifespan. Apart from text-heavy games, the joystick was clearly the dominant controller – many titles would simply refuse to work without one.

Given its commercial success and influence, the Commodore 64 has predictably been the subject of both scholarly studies and popular books. Jesper Juul examines its multiple use cases and legacy

in the *Platform Studies* series book *Too Much Fun* (2024; see also Juul & Carney 2023). The implications of the chip designs for game engines are elaborated on in Roberto Dillon's *Ready: A Commodore 64 Retrospective* (2015) and the SID chip is discussed in detail in Kenneth B. McAlpine's *Bits and Pieces* (2018, 69–102). Furthermore, there is ample literature on the cultural impact of the Commodore 64, such as on the development of the aesthetics of electronic music (Newman 2017) or the demoscene (Botz 2011; 2017). The popular history book *On the Edge: The Spectacular Rise and Fall of Commodore* (Bagnall 2005) provides a detailed account of the birth of the company, the development of many of its products, as well as the personalities involved (see also Tomczyk 1984).

Sinclair ZX Spectrum: The British Touch

The Sinclair ZX Spectrum, developed by Sinclair Research and released in 1982, occupies a significant place in the history of European home computing. Designed as an affordable home computer, its relatively inexpensive hardware and compact form made it accessible to a broad audience (e.g. Haddon 1988; Lean 2014). At the same time, the absence of dedicated graphics and sound hardware meant that most audiovisual effects had to be implemented directly in software through highly optimized programming techniques (Marecki, Yerzmyey & Straka 2020). Combined with cassette-based distribution and type-in programs published in computer magazines, this fostered a participatory culture in which users frequently modified, shared and created software themselves.

ZX Spectrum's influence extended beyond the United Kingdom into Eastern Europe, where it became particularly popular despite limited official distribution. For example, in communist Czechoslovakia, home computers spread primarily through informal networks rather than commercial marketing. By the late 1980s, the ZX Spectrum had become the country's most widespread home computer platform (Švelch 2018). Limited access to authorized servicing and peripherals also encouraged do-it-yourself repairs and hardware modifications. The popularity of the ZX Spectrum was reinforced by a large ecosystem of compatible systems and clones, including the Timex Computer 2048 and the Didaktik Gama. Due to the relative simplicity of its architecture, the ZX Spectrum became one of the most extensively cloned computer platforms in the Soviet Union and other Eastern Bloc countries, leading to a wide variety of unofficial models, such as the

Orel BK-08 (Ukraine), NPO Rotor Robik (Ukraine), Elara Delta S-02 (Russia), NTK Composite (Russia) and Signal Sintez 2 (Moldova) (Sinclair Collection n.d.; see Stachniak 2015).

The original ZX Spectrum's architecture comprised a Zilog Z80 CPU running at 3.5 MHz, ROM and RAM memory, cassette input/output, an internal speaker, video circuitry and the Ferranti ULA (Uncommitted Logic Array), which coordinated many of the machine's core functions (Smith 2010). Much of the system's software structure was inherited from the earlier ZX81 in order to reduce development costs and accelerate production, resulting in a number of technical compromises that programmers would have to work around (Adamson & Kennedy 1986).

The machine's graphics system introduced several constraints that became central to its visual identity. To conserve memory and reduce hardware complexity, the display combines a highly non-linear bitmap layer (256×192) with a separate color attribute layer organized into 32×24 cells of 8×8 pixels each (Smith 2010). There are eight basic colors, but the cell position specific brightness bit increases the number of possible options to 15. Because each cell can display only one foreground and one background color, graphics are subject to the well-known *attribute clash* or *color clash* effect when differently colored objects occupy the same cell. Although often viewed as a limitation, this design significantly reduced memory requirements and encouraged programmers to develop creative techniques, such as color reduction and dithering to minimize its visual impact. The ULA is not capable of common game-related features, such as smooth scrolling, sprites or repeating character tiles across the screen, which leaves such tasks solely to the main CPU.

The sound capabilities reflect similar hardware economies. The original 16K and 48K models lack a dedicated sound chip and instead rely on a simple internal speaker, or "beeper", controlled directly through the ULA. Audio is generated in software by rapidly switching the speaker between on and off states, requiring constant CPU involvement (Smith 2010). For this very reason, many ZX Spectrum games do not feature background music. Despite these limitations, programmers developed increasingly sophisticated audio techniques, including pulse-width modulation (PWM) and rudimentary sampled sound (see Marecki, Yerzmyey & Straka 2020; McAlpine 2018, 57–8). Later models, starting from the Spectrum 128K (1985), contain the common General Instrument AY-3-8912 sound chip, providing three independent channels and substantially expanding the machine's audio capabilities.

ZX Spectrum's original, modest rubber keyboard is one of its defining features (e.g. Reunanen, Hoeberechts & Viljanen 2024). In the absence of dedicated arrow keys, default controls were commonly mapped to Q/A for up/down and O/P for left/right, with space or M acting as the trigger. The Spectrum Plus models added arrow keys to the layout, but they were poorly positioned for games and by then the QAOP and similar conventions had already become established. The original 16K and 48K ZX Spectrum models have no built-in joystick ports, requiring an external interface for joystick input. These interfaces differ in the number of joysticks they support: the Kempston expansion accommodates one joystick, whereas Sinclair's ZX Interface 2 provides for two. The Kempston requires specific support from the software side, as the data is read from a dedicated hardware port, whereas the ZX Interface 2 maps joystick inputs to the keyboard's number keys. Thus, redefinable keys allow games without explicit support to work with a joystick (Sinclair Collection n.d.).

The ZX Spectrum has also received considerable attention within the history of computing, although no dedicated volume in the *Platform Studies* series has yet been published. Existing scholarship has approached the platform from several complementary perspectives. Leslie Haddon (1988) examines the emergence of the home computer as a consumer technology and places the Spectrum within the broader development of the domestic computing market (see also Lean 2014). Jaroslav Švelch (2018) explores the platform's adoption in communist Czechoslovakia, emphasizing informal distribution networks, amateur programming and user practices behind the Iron Curtain. Zbigniew Stachniak (2015) extends this perspective by outlining the proliferation of unofficial ZX Spectrum clones across the Soviet Union and their role in the development of local computer hobby cultures. From a technical perspective, Smith (2010) provides one of the most comprehensive analyses of the Spectrum's hardware architecture and design.

MSX: Competing, yet Collaborating

Unlike the other two contenders, the MSX is not a computer model by a single company, but a standard of mutually compatible 8-bit machines. After its introduction in 1983 by the ASCII Corporation in collaboration with Microsoft, several companies started producing their own models that shared the same underlying hardware, yet looked physically very different from each other. In addition to Japanese electronics giants, such as Sony, Canon, Panasonic and Toshiba, also

foreign enterprises tried their hand in the market. Some examples of them were South Korean Daewoo and GoldStar, American Spectravideo, Dutch Philips and Brazilian Gradiente.

The original MSX hardware specifications were clearly inspired by, but not compatible with, the Spectravideo SV-318 and SV-328 computers. The design of the MSX was safe, based on tried and tested off-the-shelf components: the Zilog Z80 CPU (3.58 MHz), Texas Instruments TMS9918 video display controller (VDP), and three-channel General Instrument AY-3-8910 sound chip (PSG) or its Yamaha-produced clone. Some other contemporary competitors, such as the British Memotech and Tatung Einstein, also featured almost identical hardware, but were not compatible with the MSX. The standard was later augmented as the MSX2 (1985), MSX2+ (1988) and Panasonic's Turbo R (1990). Most of the improvements concerned video capabilities – resolution, number of colors and scrolling – but depending on the model, also faster CPU speeds, FM audio synthesis and more memory could be found.

From a programmer's perspective, one of the defining features is the TMS9918 VDP, which, unlike most comparable solutions of the time, has its own dedicated 16 kilobyte video memory (VRAM). The graphics are essentially character-block-based with a maximum resolution of 256×192 pixels and 15 colors. On the MSX the VRAM is accessed through so-called port commands instead of direct memory addressing, which makes random access and modifying existing content cumbersome. Even if the VDP supports as many as 32 single-color 16×16 pixel sprites, only four of them can be visible on the same line, which limits their usefulness; the ones with lowest priority are not displayed on lines where there are five or more present. There was no smooth scrolling capacity before the MSX2, leading to coarse eight pixel jumps in scrolling games.

The VDP is capable of four graphics modes, out of which SCREEN 0 (as per its BASIC name) is a two-color text mode. SCREEN 3 is a coarse 64×48 freely addressable pixel mode, whereas SCREEN 1 could be called a text/graphics hybrid with a joint redefinable character set for the whole screen – game screens could be constructed out of the tiles. SCREEN 2, often used for games, allows for most precision with its 32 by 24 blocks that can be individually redefined for each location on the screen. SCREEN 2 can also be thought of as a 256×192 pixel mode where eight neighboring pixels can be freely chosen from two colors. The *pattern table* contains pixels for the characters as bits, the *color table* their color information and, finally, the *name table* decides which tile is shown at each location. Because of the need to address every character position independently, the screen is divided into independent horizontal thirds that each have their own character set.

Joystick ports are a standard feature on MSX computers and, rather exceptionally, the hardware supports two separate buttons. In practice, most joysticks of the time only featured one button; even if more were present, they were wired to the same output. Japanese Konami games, which represented the peak of MSX gaming in many respects, supported both the keyboard and joystick. Keyboards and, in particular, the arrow keys used for gaming vary greatly across different models; at times they are simply normal keys in the common inverted T layout, but also separate larger keys exist in different configurations and colors. The Canon V-20 is an extreme example with its drumpad-like design: the up and down keys are both 7cm wide. This diversity meant that the tactile experience was far from universal for MSX gamers, unlike on platforms with a single manufacturer.

As to academic studies concerning the MSX, there is little available as of now. The platform is mentioned in several studies dealing with the history of computing (e.g. Veraart 2008; Oliveira et al. 2014; Suzuki 2016; Yamashita 2020), but in-depth accounts still remain to be written. The *Platform Studies* series has not taken up the MSX yet, either. Outside academia, there is no shortage of literature, as the hardware has been documented in depth (ASCII Corporation 1984; Avalon Software 1985) and there are numerous popular books on, for example, various programming topics, general system use and games (e.g. S.T.A.R 2012).

From the 1980s to Today

The heyday of 8-bit home computers seems firmly set in the 1980s. By the mid-1990s, they had effectively disappeared from retail stores. Companies like Commodore folded, whereas Amstrad would eventually no longer profit from the Spectrum range or their own CPC machines. 8-bit game magazines and game releases ceased shortly after, but not instantly. For example, Amstrad produced the last ZX Spectrums in 1992, while the magazine *Your Sinclair* published its final issue in September 1993.

The original commercial run of a home computer platform can be seen as completely finished after the computer hardware was no longer produced and the mainstream game releases and newsstand magazines ceased. This also marks the starting point for considering a platform's afterlife, which may now span multiple decades. Jesper Juul lists no less than five different ”lives” of the Commodore 64. Of these, demoscene and hobbyist projects have certainly extended long

beyond the original commercial era, while the visual and aural styles associated with the computer no longer seem merely nostalgic, but as a deliberate choice in art or music. (Juul 2024, 1–2.)

In the 1990s, PC compatibles and Commodore Amiga computers became powerful enough to emulate 8-bit computers. This is one milestone in the story of 8-bit computers, as new hardware could play the part of older platforms without the need to convert existing software (for more on emulation, see Woods & Brown 2008; Newman 2012; Ippolito 2016). As emulation matured in the 2000s, emulator software became the target environment for developing software for the old platforms. The threshold for cross-developing games became lower, and hobbyists could continue producing games and other software without resorting to a cumbersome 8-bit hardware setup or without owning the actual target hardware at all. Thus, the role of the original platform increasingly turned toward a technical reference point, as opposed to a practical everyday tool.

As nostalgia items, products related to the original platforms have also enjoyed a commercial resurgence during the 2010s and 2020s. Various recreations of the original computers can again be found from home electronics stores (e.g. The C64 Mini, C64 Handheld, The Spectrum and Sinclair ZX Spectrum Vega). These gadgets are often intended as low-threshold products for playing a built-in selection of memorable games. The computers are typically not based on the original hardware, but either an FPGA simulation, or an emulator running on a lightweight Linux distribution or an embedded system. These products often lack original video connectors, instead opting for a more modern HDMI that conveniently plugs into today's televisions. Similarly, modern gamepads can be connected via USB, whereas the old Atari compatible 9-pin joysticks cannot be directly used. In recent times, the community has also seen the "official" re-release of the C-64 (Commodore 64 Ultimate), which is highly compatible with original peripherals.

Numerous websites document 8-bit computers and provide software from the original commercial era, often omitting copyright concerns. Discussion groups first emerged on internet forums and later on social media, dedicated for revisiting old games or presenting new demos and games. A new kind of symbiosis has emerged between indie game and magazine publishers, a cottage industry of hardware, replacement parts, emulators together with their internet audiences and participants. Hardware enthusiasts provide new expansions to old platforms, for example enabling fast loading of software from memory cards. The activity and networking between enthusiasts on the internet can be seen as the glue that keeps the 8-bit afterlife thriving.

In 2026, the afterlives of 1980s' computers are already much longer than their commercial lifespans were in their heyday. All three platforms of our study are well alive with new games, hardware expansions and even complete system remakes produced on a regular basis. To consider such platforms as historical artifacts alone would be ignorant, as they have, in many respects, made a comeback and returned to people's lives – not as the everyday workhorse any more, but an active hobby instead (cf. Lindsay 2003).

Three Games

Like with the platforms, the choice of games is a crucial decision, as it will heavily dictate the outcome of the analysis. A poor selection will not highlight the capabilities of each machine or the creative solutions that programmers, graphics artists and musicians applied in order to create the best possible outcome. Focusing only on games requires some justification, too, as they certainly were not the only kind of software that was developed and used at the time. Our decision here is based on the fact that action games, in particular, utilize the target hardware widely: there is music, sound effects, static and moving graphics, as well as interaction using a game controller or the keyboard. The real-time nature of such games often set considerable demands for the efficiency of the program code. In contrast, analyzing a text editor would reveal much less about the topic.

Before moving any further, it is necessary to ask what constitutes a "game" to begin with. As pointed out multiple times by James Newman in his *Best Before: Videogames, Supersession and Obsolescence* (2012), a single, definitive game does not even exist. For example, regional and platform-specific versions, updates, bug fixes and re-releases make such definitions trickier when talking about a specific title. Especially on home computers, software often spread as pirate versions that were not necessarily equal to the official release (see also Vuorinen 2007; Wasiak 2012). In the same vein, we are dealing here with games that originated on one platform and were later ported or converted to several others (cf. Fernández-Vara & Montfort 2013).

When is a game version a port, conversion, remake, clone or perhaps a remaster? Grabarczyk and Aarseth (2019) discuss these concepts at length and present an ontology to clear up some common misconceptions. For the purposes of this article, we take into account their main distinction, the difference between a port and conversion: a *port* is at least partially based on original code and assets, whereas a *conversion* relies on replicating the original functionality and

look independently – a *remake* is just an instance of the latter. In reality, it might be impossible to tell which one we are dealing with, as the production process of a particular game is so often opaque to outsiders.

Our selection process was guided by the strive for balanced diversity. Thus, the selected titles represent both different countries of origin (Japan, the US and UK) and platform origins (arcade, Atari 8-bit/C-64 and the Spectrum). At the same time, we wanted to focus on widely converted games, leaving room for future work that might address the same games on even more platforms. While it was easy to find titles that were originally released for the C-64 or Spectrum and later ported to the other two platforms, finding a similar MSX game turned out to be so difficult that an arcade game had to be chosen instead. This observation alone tells something about the dynamics of the industry. Well-known games of Japanese origin running on European or American machines were commonly conversions of high-profile coin-ops, so the choice can be considered as representative rather than a shortcoming.

The specific choice of the three games was made to represent the diverse hardware features of the platforms when performing similar tasks. A game may benefit from hardware multiplexed sprites, a characteristic feature of the Commodore 64, whereas the ZX Spectrum has to do without hardware sprites at all. The scope of game genres is focused rather than broad: each of the selected games features a side perspective and has a number of small moving objects on top of a static flip-screen or a scrolling background. A relatively similar effort, such as *Starquake* and *Bruce Lee*, can reveal differences in the details of their development for different platforms, providing rich material for study. Examining wireframe 3D graphics, isometric perspective games and third-person driving games on the three platforms could unearth further differences and points of comparison.

To study the games we have primarily used emulators running on modern PC hardware: *VICE* 3.6.1, *Fuse* 1.6.0 and *openMSX* 17.0. YouTube playthrough videos were useful in discovering game content, as it may be difficult to finish the games (cf. Therrien 2012). The purpose here has not been to become adept at playing the chosen games, but to gather sufficient experience for comparison. Emulators and videos also provide tools for rewinding and pausing, as well as capturing screenshots, which are all useful for examining the visuals in more detail. The former also facilitate deeper technical analyses, as they allow users to save complete system states for later use, as well as examine memory contents and various chip states when necessary.

The three games were also tried on original hardware connected to an original 1980s CRT display. This was mostly arranged so that we could verify if emulation and videos were accurate representations of how the games appear and how they play with original controllers. For example, quickly flickering on-screen elements may become amplified or disappear on a modern display, whereas current keyboards and gamepads are often more precise than their 8-bit home computer counterparts.

Such an exercise is not meant to replicate past conditions of play fully (cf. Newman 2012, 140–54). Such an undertaking would not be possible even in a technical sense. For example, game controllers and keyboards may no longer perform as they originally did after 40–50 years. In contrast with game consoles, 1980s' home computer platforms did not have established standard controllers. The same games could be played with a wide variety of third party joysticks, ranging from a minimal handheld stick to a suction-cup and auto-fire enhanced bulky box resembling a jet fighter stick. This flexibility also helped to level the physical playing experience across platforms, as players could own the same joystick models. Regardless of appearance, the joysticks always provided four non-analog directions and a fire button or two. Analog paddles were supported on the Commodore 64, but virtually unknown on the other platforms.

Playing on a CRT display with composite video is accurate to the period, but many users also connected their computer to a family television or a second-hand TV, using RF/Antenna cables that further blurred the image. Conditions of play were not always neat and dedicated, but situated unergonomically at a corner of a desk or on the floor of the family living room (Reunanen, Saarikoski & Joelsson 2025). Black-and-white televisions for playing games were not unheard of in the 1980s, leading to rather different experiences (ibid.).

Apart from technical details, game cultures and the reception of games have also shifted. For example, it has been pointed out that gamers in the past would engage in justification and defiance when faced with outsiders' judgment and stigmatization (Kirkpatrick 2016). Hit games of the time may appear strange or uninteresting today; impressive technical feats were, likewise, surpassed later on. Home computer games were not designed to be completed in the same sense as today, which may have resulted from inheriting arcade/coin-op designs that were meant as short-term enjoyment, rewarding highly skilled play with longer play time. Games also had relatively little content and a more difficult title could be seen as providing more value for money to the buying customer.

Starquake

Home computer games gradually began to distance themselves from the template provided by coin-operated arcade games over the 1980s (specifically on the C-64, Juul 2024, 77–8). The player at home would have more time at their hands and could also be after different experiences than at the arcade hall. Both the public and private context steered gaming to their own direction which, in its part, facilitated the diversification of game genres (cf. Huhtamo 2016; Wade 2016). Strategy, simulations and text adventures were some of the genres that could not be found in arcades, but home computer action games also expanded into various hybrids, with added exploration and puzzles.

Starquake was originally published by the British company Bubble Bus in 1985 and other versions appeared the following year (Figure 2). At this point the UK had already established itself as the home of the leading European computer game industry, whereas many of their local home computer manufacturers – Sinclair included – were facing severe economic troubles (Kerr 2012; Wade 2016; Reunanen, Hoeberechts & Viljanen 2024). The game is credited to a single person, Stephen Crow. Such a feat was still common for the time, as filling 48 kilobytes of memory with game content for the ZX Spectrum was not an impossible task for a sole developer.

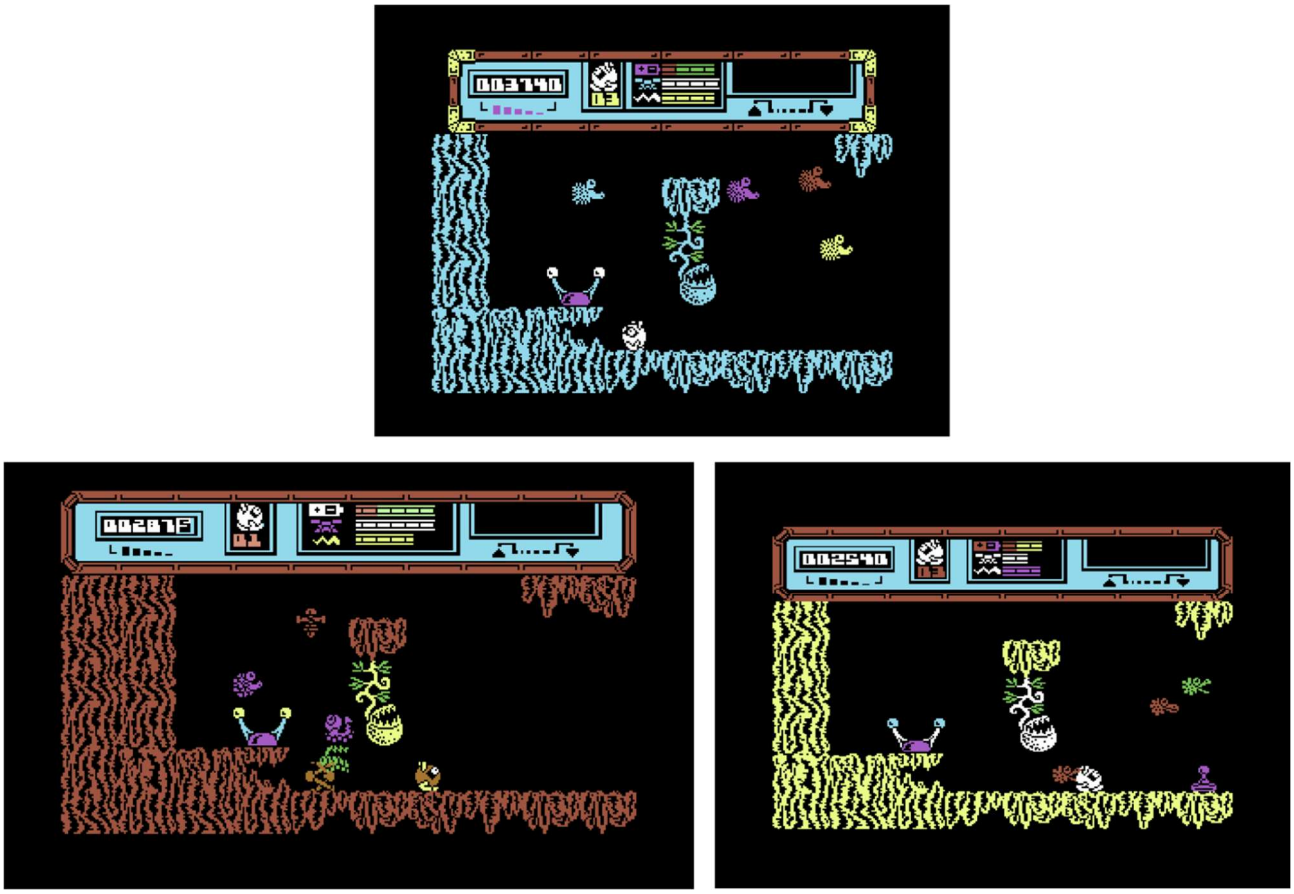


Figure 2. Starquake running on the ZX Spectrum, Commodore 64 and MSX showing the same room.

Starquake is what was back then called an ”arcade adventure” in the British games press. It is an action game with conventions borrowed from text adventure type games, with a maze of rooms and items carried in an inventory. It appears as an imitation of Ultimate’s earlier hits, such as *Atic Atac* (1983), *Sabre Wulf* (1984) and *Underwurlde* (1984). *Starquake* attempts to clear the bar set by Ultimate with its colorful cartoon style graphics and fast motion. The world has 512 screens – a large map size like this was a marketable feature against what could have been seen as simplistic single-screeners. In its genre, *Starquake* is particularly hectic: the player moves fast among tiny critters that appear randomly and rush around. Despite the adventure leanings there is no save function; a password-based teleport system helps the player to advance in the game. Game magazines of the time readily published maps, tips and cheat patches to help complete these difficult arcade adventures.

Starquake was very widely converted (C-64, MSX, CPC, BBC, Tatung, PC, Atari ST and Atari 8-bit). This pluralism was helped by a design that does not use the ZX Spectrum capabilities in any way that would have been difficult to replicate on other platforms. The player character is 16×16 pixels big while the moving critters employ 24×16 pixels. This is easy to manage with hardware sprites on

other platforms, and the game does not attempt scrolling or complex graphical feats. Flip screen games were especially suitable for the Spectrum, as major screen redrawing, such as scrolling, would produce slowdowns.

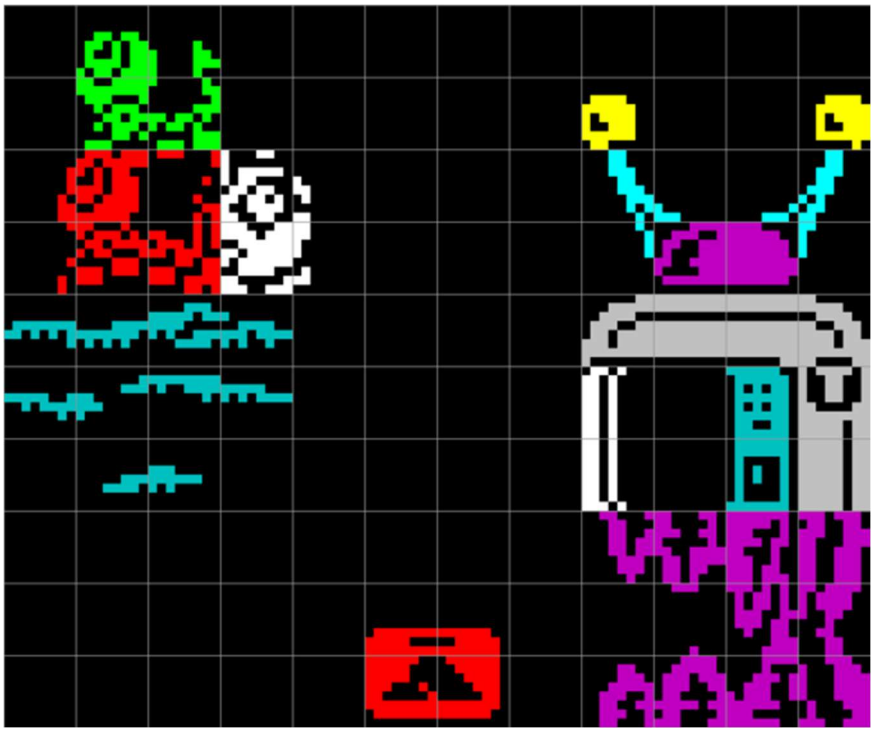


Figure 3. A cropped screenshot of Starquake with a grid to highlight the ZX Spectrum color limitations. Static background elements safely align with the grid, while the moving objects at top left compete for the color space, producing an attribute clash.

Moving elements mainly consist of the round protagonist called BLOB and its various enemies. They are all single-color to enable pixel scale motion and animation, and to avoid attribute clash, i.e. a visible artifact resulting from two differently colored moving objects competing for the lower resolution color space (Figure 3). This could be avoided by favoring static background elements and easily manageable color combinations, such as colors on top of black background, a common approach to compromise between color use and clash (for examples, see *Sinclair ZX Spectrum: A Visual Compendium*, Jarratt 2015).

As the ZX Spectrum has no hardware sprites, any sprite-like moving elements, ”software sprites”, are achieved by altering the memory-mapped display contents directly. Unlike with hardware sprites, changes to the screen also need to be reverted or cleaned, again expending processor time. ZX Spectrum software used a number of approaches ranging from primitive, flickering direct drawing operations to full outline-masked, screen-synchronized, double-buffered animated

objects. *Dan Dare* (1986) and *Deviants* (1987) had moving elements partly hidden by the block-aligned scenery, a neat solution that adds visual variety while reducing color clash.

Dk'tronics' *Popeye* (1985) and *Savage* (1988) from Firebird Software have very large objects that move in 8-pixel increments, again avoiding the clash almost entirely. *Starquake* has an empty background, so no separate masking operations or further tricks are to be found. Pre-shifted animation frames, which use more memory but are faster to copy to the byte-aligned screen, are utilized in the game (see Figure 4). The result is recognizably a "sprite" for anyone familiar with the Commodore 64 or arcade games.

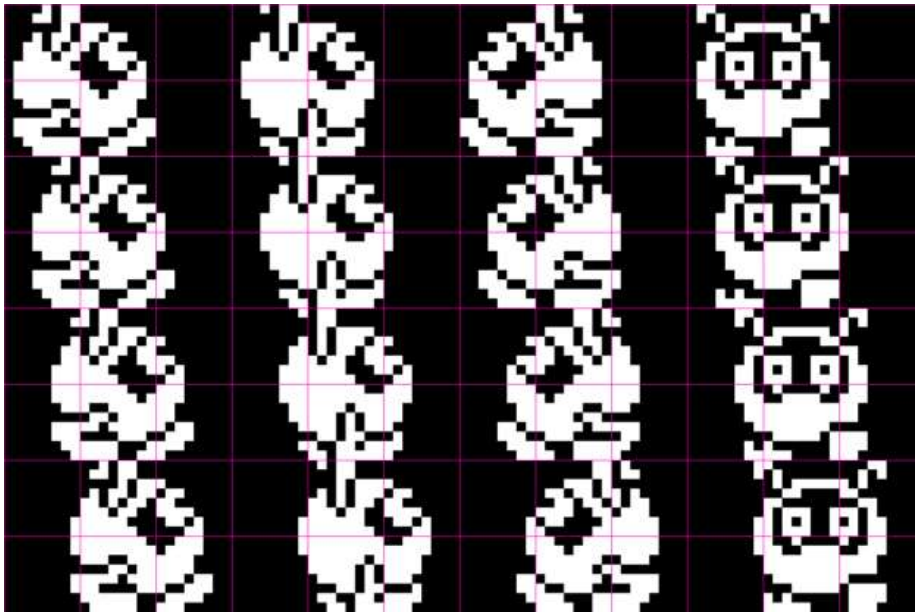


Figure 4. An excerpt from the memory contents of the ZX Spectrum version of Starquake shows 24×16 pre-shifted variants of the same animation frames, which facilitate faster display of a 16×16 visual element.

The European MSX game range suffered from direct ZX Spectrum ports. This was an understandable approach, as the CPU was the same and the MSX could match its resolution and colors. The graphics routines were likely to be just adapted directly for the VDP chip instead of rewritten in their entirety. For example, the MSX version of *Chase H.Q.* (1988) looks nearly identical to the Spectrum one, but runs slower, owing to the different port and character-based video memory access. In addition to the speed issues, the underuse of the MSX's color, sprite and sound capabilities are other staples of low-effort ports. In contrast, *Starquake* has been converted more meticulously, making it a richer case for comparison.

The MSX version uses hardware sprites to avoid color clash and slightly upgrades the sound from the beeps of the original. Interestingly, there is no evidence of disappearing or flickering sprites, as their number on the same line stays below five most of the time. This is hardly just a coincidence, but rather a conscious design decision made early on. While the Spectrum and C-64 versions feature very fast screen redraws when entering a new room, the MSX version visibly lays down the screen tiles, adding little pauses to otherwise rapid proceedings. On the one hand, the slowness follows from the port-based video memory access and, on the other hand, from lack of platform-specific optimization, as it is technically possible to refresh the whole 16 kilobytes of VRAM in about 0.1 seconds. A peek into the name table reveals that *Starquake* does not repeat identical tiles across the screen, but treats it as a Spectrum-like bitmap. The decision has made the porting simpler, yet failed to utilize MSX hardware capabilities optimally.

As the C-64 offers a high resolution mode with ZX Spectrum-like color limitations, converted titles could often look almost identical. Spectrum originals, such as *Lords of Midnight* (1984) or *Laser Squad* (1988) appear very similar on the C-64, despite making use of the wider screen resolution. The Commodore 64 version of *Starquake*, likewise, has a wider play area, adapting the rooms with added filler to the left and right (Figure 2). Multiple sprites are used for a colorful, yet high resolution main character. Again, the original attribute clash is avoided. Despite all this, the display is dominated by the black background approach, which still strongly refers to the ZX Spectrum, compared to many C-64 originals with their metallic grayscales in the 160×200 multicolor mode.

Owing to the CPU intensive sound playback, the original *Starquake* does not feature background music: there are only simple sound effects. However, more effort has been invested into the short title screen tune, which showcases the possibilities of the ULA when the processor can be fully dedicated to audio synthesis (cf. Marecki, Yerzmyey & Straka 2020). On the C-64, the atmosphere benefits from SID-enabled sounds that include echo to make them livelier. Curiously, the plain click-clack walking sound of the original has been replicated as is, even if the chip would facilitate more refined effects. The MSX version, in turn, hardly utilizes the possibilities offered by the PSG, as the audio closely resembles the Spectrum one, albeit with less crackling and a different walking sound.

Starquake highlights certain aspects of the game conversion industry particularly well. Firstly, the ZX Spectrum can be thought of as the ”lowest common denominator” that most other machines could match in both audiovisual capabilities and processor speed. Starting from that premise, it

was almost certain that a manageable port of a Spectrum original was doable. The problem – at least from the point of view of other platform owners – was rather that the extra features present on more capable devices would go underutilized, leading to colorless and slow versions with only rudimentary audio. Owing to the popularity of the Sinclair machine in the UK, several titles ended up being exactly like that; *Starquake* here exhibits some of the symptoms as well, but it certainly is not among the worst examples.

Gradius/Nemesis

Nemesis (1985) – originally called *Gradius* in Japan – is a milestone in the 1980s horizontally scrolling shooter genre (Figure 5). It extends the ideas previously found in *Scramble* (1981) and *Defender* (1981), adding a complex power-up system that radically expands the play. The player gathers power-up symbols, enabling *speedup*, *missile*, *multi/option*, *laser* and *shield*. *Nemesis* is an arcade conversion, made by three different parties in three notably different ways. Konami produced the original arcade game and the MSX conversion (see S.T.A.R 2012). The ZX Spectrum version (1987) was developed in the UK by Cyclone and the C-64 (1986), likewise, in the UK by Simon Pick. All three were first published under the Konami label. Typical for the company, the MSX *Nemesis* is a ROM cartridge, which greatly speeds up loading times and provides extra space compared to the single-load tape and disk releases of the two other platforms; fighting piracy was likely another motivation behind the format.

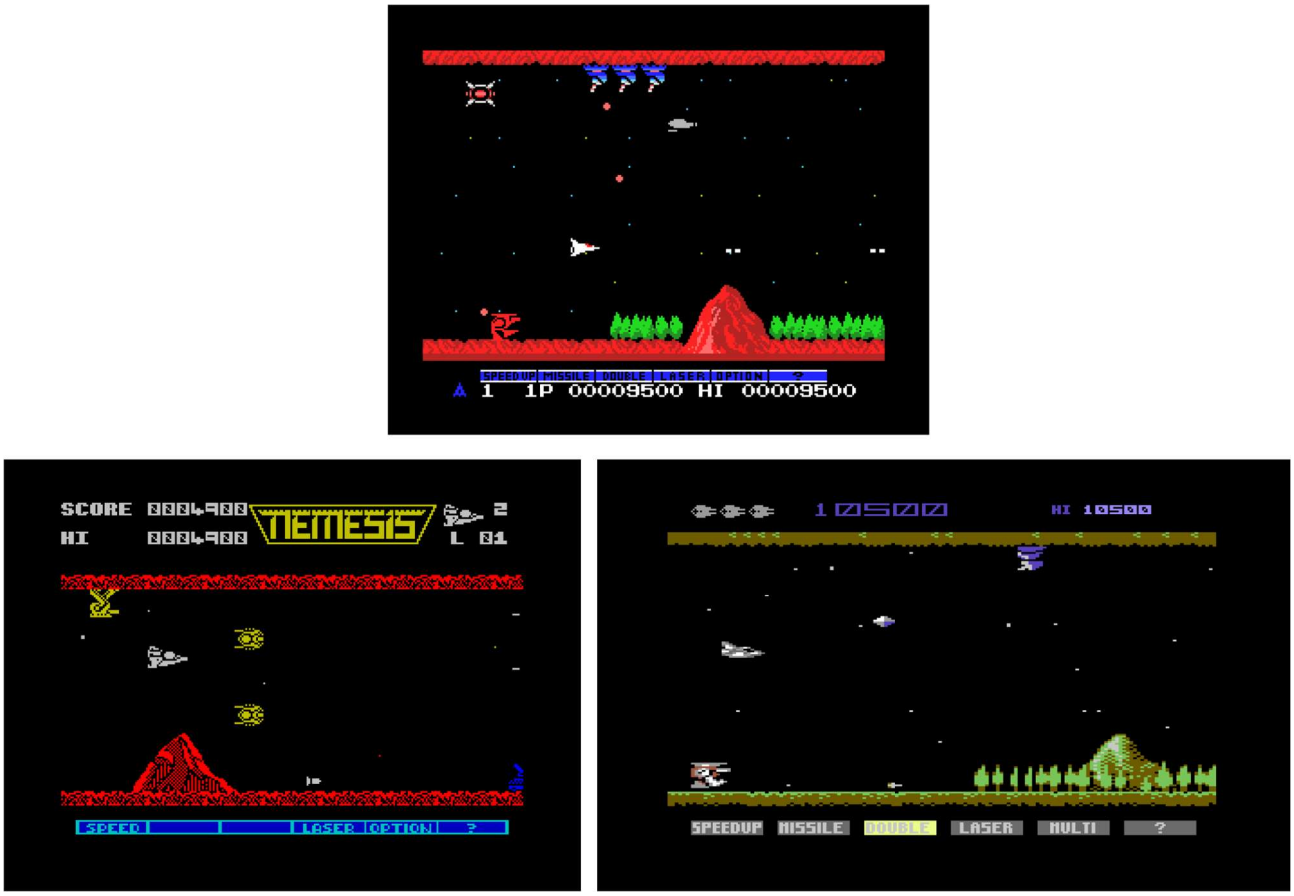


Figure 5. Nemesis running on the MSX, ZX Spectrum and Commodore 64.

The early European home computer era tended to produce knock-offs of known arcade games, such as *Pac-Man* (1980) or *Space Invaders* (1978). Later, official licensing became commonplace, even if blatant clones never disappeared from the market, either. Japan was a hotbed of arcade games in the 1980s, and several popular titles flowed from there to other continents as cabinets or computer/console conversions. Coin-ops that were known from amusement arcades provided some free advertising for home versions. The setting where modest video game consoles and home computers looked up to arcade originals running on much more powerful, dedicated hardware remained in effect well into the 1990s: to call a conversion ”arcade perfect” was high praise for the time (see Grabarczyk & Aarseth 2019).

Differences between the three versions discussed here are instantly apparent. It takes further examination to find out which of them are due to the properties of the hardware platforms, and which originate more from the skills and resources of the authors behind the conversions. Several necessary compromises have been made in the content, especially in the ZX Spectrum one. All versions replicate the Konami power meter system, probably seen as a defining feature of the game. Notably, the ZX Spectrum version only allows one *option*, an ethereal copy of the player ship

which follows its movements and multiplies all other weapons. An interesting detail relates to the controls: as the Atari joystick supports only one button, the power-ups have to be activated from the keyboard, which can prove fatal during hectic gameplay – an MSX-specific two-button joystick facilitates safer selection.

The Commodore 64 has been described as the ideal scrolling machine (Juul 2024, 93–6). This feature is not simple to program, as it is not possible to simply point an offset to an already existing larger background image. The display is shifted pixel-wise by changing the contents of a video register, allowing values of 0–7. The software then needs to move or redraw screen contents before the cycle is repeated. As smooth scrolling was non-trivial to implement, games that made use of scrolling could be appreciated for technical execution and skill. The C-64 *Nemesis* only needs to scroll the level from right to left at constant speed, which makes it easier to program than, for example, an 8-way scrolling background. The tile map is multicolored and is varied in complexity.

Early C-64 games used sprites almost exclusively for players and missiles, and characters for backgrounds, whereas later developers used them more creatively, basing entire game ideas more on the strengths of the computer (Juul 2024, 100–7). In later games, a large moving enemy might be constructed out of characters and some background elements could be made of sprites, an insight that one developer credits to demoscene habits (Trenz in Juul 2024, 103–4). In this trajectory, the *Nemesis* conversion could be described as a middle-era game, as sprites are mostly reserved for moving objects, but also characters are used for missiles and some of the enemies.

The arcade original serves as a difficult target, presenting ideas that might not have appeared had the game been a C-64 original. This complex content separates it from games like *Rambo* (1986) and *Parallax* (1986), which showcased technically impressive scrolling but somewhat simplistic play on top of these sophisticated backdrops. To handle all the sprites, *Nemesis* needs to employ a multiplexer routine, which allows the hardware to re-use the eight sprites as long as the same sprite number does not occupy the same horizontal area. Even then there is sprite flicker, something that could not even be alleviated by slower CRT refresh times. This technical shortcoming, alongside the use of blocky extended sprites was already pointed out in contemporary reviews (e.g. *Zzap!64* May 1987, 90–1).

The ZX Spectrum version reduces the scrolling area more heavily than the other versions, which was a common approach in Spectrum games. This reduction made the needed copying and

redrawing less taxing on the CPU. The scrolling routine is not especially effective, and tricks have been used to make it appear more comprehensive than what it actually is. Unlike in other versions, there does not seem to be an underlying tile map for the levels – the hardware does not support such in any case. The levels are constructed from sprite-like moving elements which are separate from each other, while a narrow band at the top and bottom of the play area adds to the impression of full screen scrolling.

As the ZX Spectrum cannot rely on hardware sprites or scrolling, the game speed has more to do with the overall drawing budget, which is managed by the aforementioned tricks. The machine could draw horizontal laser beams very effectively (as evidenced by *Jetpac* [1983]), but very little of that is visible in *Nemesis*. Apart from missing game content and occasional slowdowns, glitches in bullet movement and collision detection reveal a patchy conversion; in general, there is a feeling of emptiness with little action on the screen.

The same asceticism is repeated in the audio domain: while the C-64 and MSX players can enjoy background music and sound effects (selectable on the C-64), on the Spectrum there are only simple blips, apart from the title screen, where a rough version of the *Nemesis* tune plays in the background. By 1987 the ZX Spectrum 128K with its improved sound chip had already been two years in the market, but the game does not utilize such extras. Possibly due to the spacious cartridge format, the MSX version features different background music for every level, unlike on the C-64, where the same tune repeats everywhere. The C-64 and ZX Spectrum versions of *Nemesis* are single files loaded from disk or tape, implying a modest production goal. For example, the later ZX Spectrum conversion of *R-Type* (1988) made use of in-game file loading (”multiload”) and included more elements from the arcade original.

The MSX version immediately reveals blocky scrolling that advances in 8-pixel leaps, in correspondence with the underlying character blocks. However, plenty of attention has been paid to replicate content from the arcade original, which must have been more readily available to Konami’s own developers. The MSX sprite capabilities are relatively well suited for *Nemesis*, as enemy patterns and objects tend to stay in their horizontal lanes. Horizontal player bullets are drawn using characters, whereas the diagonally moving missiles are hardware sprites. The enemies that do not need to move at pixel accuracy are character blocks, removing the need to render them as sprites (Figure 6). Despite the hard limits in sprite use, two of them are used for the player

ship to provide a spot of color. Cleverly, as the blue underside is revealed, the red cockpit is not visible and vice versa, avoiding the need to use three sprites.



Figure 6. The MSX version of Nemesis with sprites circled. All other graphics consist of characters.

As the number of on-screen objects grows larger, especially when employing multiple *options* or in the crowded volcano scene at the end of the first world, the rendering routines are not entirely able to keep up. Because of the MSX sprite hardware the overload does not produce flicker, but results in clipped-off and disappearing sprites when more than four of them occupy the same line. The background graphics and many enemies are more colorful than in either of the two other versions, utilizing the large color memory and flexible color placement (two colors for a 1×8 pixel region, see Figure 6). In contrast with the blocky sprite-based end of level boss in the C-64 version, the large alien ship is colorful and precise. All in all, the MSX version is well thought out, with a purposeful blend of character and sprite graphics that takes hardware features well into account.

From a technical perspective, *Nemesis* is the most challenging title of our selection, as it was initially developed for dedicated arcade hardware whose features considerably surpassed those of domestic platforms of the time. The conversions were inevitably playing catch-up when trying to replicate the colorful, snappy action of the original. Sprites and character tiles were available on the MSX and C-64, and the latter would also benefit from its scrolling capabilities, whereas the ZX Spectrum had none of these features, making a good quality conversion an almost impossible goal. In the big picture, *Nemesis* also highlights the dynamics and practices of the contemporary game

industry, as well as the unbalanced relationship between coin-ops and home computers, which remained largely unchanged until the 1990s.

Bruce Lee

Bruce Lee from 1984 could be categorized as a platformer or a fighting game (Figure 7). The plot follows the titular martial artist as he progresses from chamber to chamber, avoiding traps and collecting lanterns to open doors. The main opponents that appear on many screens are the wooden sword wielding *Ninja* and the sumo wrestler lookalike *Green Yamo*. *Bruce Lee* came out at a time when martial arts cinema was highly popular in the West; in spite of its name, the game does not resemble any actual Bruce Lee movie, although at least *Enter the Dragon* (1973) must have inspired it to some extent (cf. Hunt 2003; Morris 2004).

The game was first written by Ron J. Fortier for the Atari 8-bit line of computers, but the C-64 and Apple II versions followed shortly thereafter – the machines are all based on the same MOS 6502 processor, and the C-64 version is effectively identical to the Atari one due to their similar hardware features. Hardware sprites are used for the moving combatants, while the character display mode provides a tile-based backdrop for the play, yielding a highly suitable combination for the C-64. The Apple II version stands out from the two others with its distinctive colors, low resolution and software-based sprites.

Bruce Lee was originally published by Datasoft, an American company, and versions for the European market, most notably the Spectrum and Amstrad CPC, were done by the British company U.S. Gold, known for importing American games to Europe. Like the Spectrum and CPC, the MSX is based on the Z80 processor, so original code could not be used as is which, in its part, led to notable differences between the versions, coupled with considerably different underlying hardware capabilities. Multiple other conversions exist, out of which the BBC Micro version is a curious – borderline absurd – outlier with its purple-skinned and yellow-haired Bruce, missing Ninja and mindlessly rushing Green Yamo.

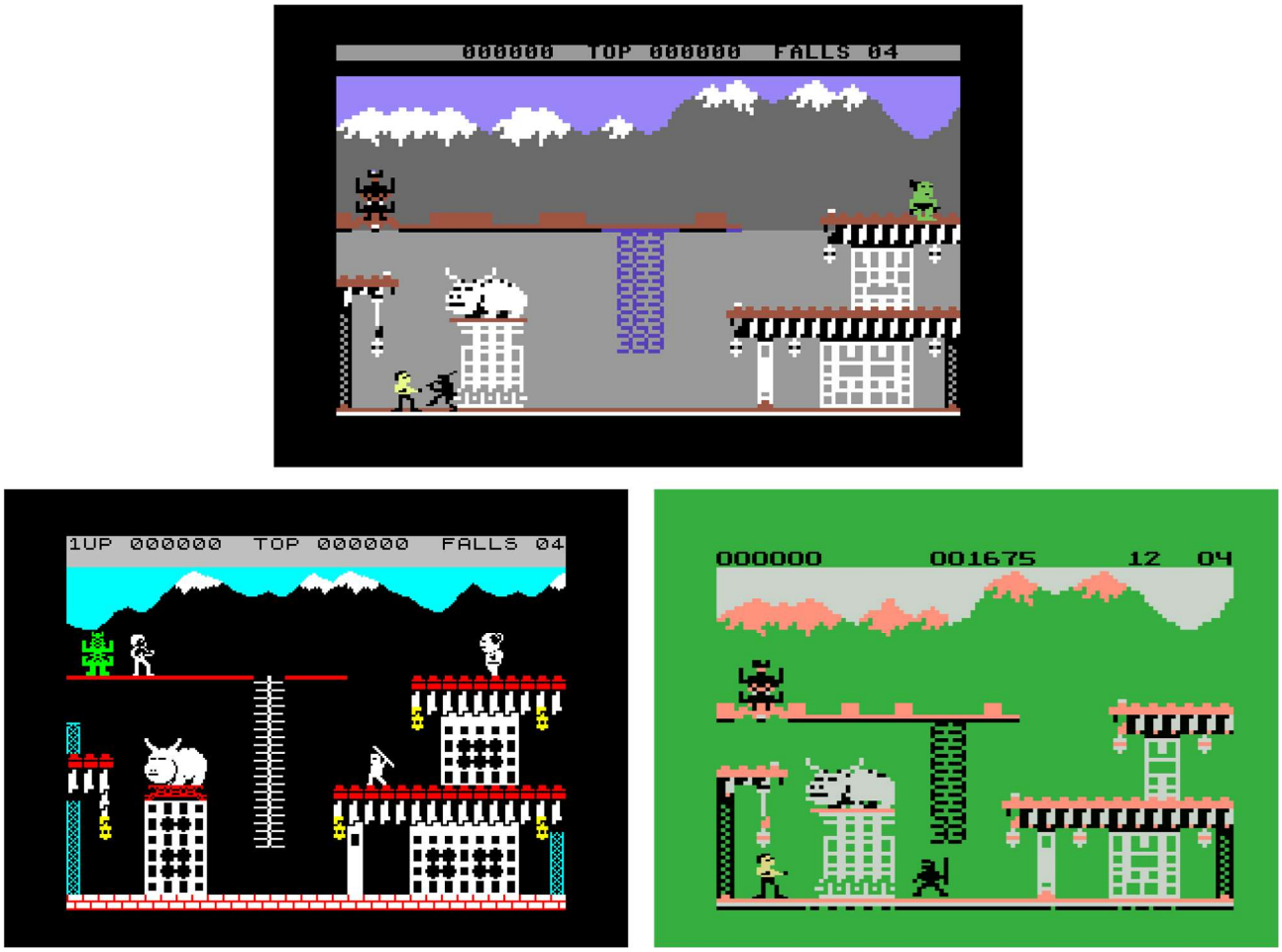


Figure 7. Bruce Lee running on the Commodore 64, ZX Spectrum and MSX.

The MSX version, published by Japanese Comptiq in 1985, appears mostly faithful to the original, apart from the missing title screen and score labels, plus the exceptional background colors. For instance, in the first screen the screen border and background including the mountains are green, which is hard to explain, as no other version looks like that (Figure 7). Running on a PAL machine, the image appears considerably wider than on the NTSC. In theory, the MSX VDP has plenty of sprites to handle the game characters, but as Bruce and Yamo both require two colors, even the single-color Ninja is too much when all three are on the same floor. Bruce's jump kick, too, requires two horizontal sprites, so the characters flicker plenty and at times parts of them disappear. One more undesirable feature is the slow falling speed that makes it harder to avoid enemies and traps – minute differences like this give rise to considerably different gaming experiences across platforms.

The ZX Spectrum has most to adapt to: the program crosses over from the 6502 to Z80, there are no hardware sprites and the screen is a bitmap. This version uses a black background which, as

already mentioned, was common in Spectrum games. The black color often had a transitional function, as there are two brightnesses for all the other colors but black. Therefore, a background element on top of black could safely use two different brightness levels. This idea is not put into maximum use in *Bruce Lee*, as the visuals are uniformly bright.

The static nature of the game screens with only three small moving characters and little other animation has made the Spectrum version straightforward to make, when compared to action-packed titles with scrolling and more extensive animation. Still, it is worth noting that while the C-64 or MSX could update single tile contents directly to animate the large moving escalators with virtually no extra cost to the processor, the Spectrum has to make all such visible modifications to the screen bitmap byte-by-byte, expending more CPU cycles.

The lack of color definition that was available in the C-64 and MSX hardware sprites meant the ZX Spectrum figures of this size almost necessarily needed to be adapted to 1-bit single color format. Masking, another common ZX Spectrum software sprite technique, makes the characters opaque and gives them an additional outline, which prevents them from getting mixed with the background. The Bruce, Yamo and Ninja characters are curiously mismatched, as Bruce is literally black and Ninja is white (see Figure 8). Yamo appears most balanced for the black background color and the masking, whereas the other character designs would work better on some other color scheme, possibly black pixels on a light background.

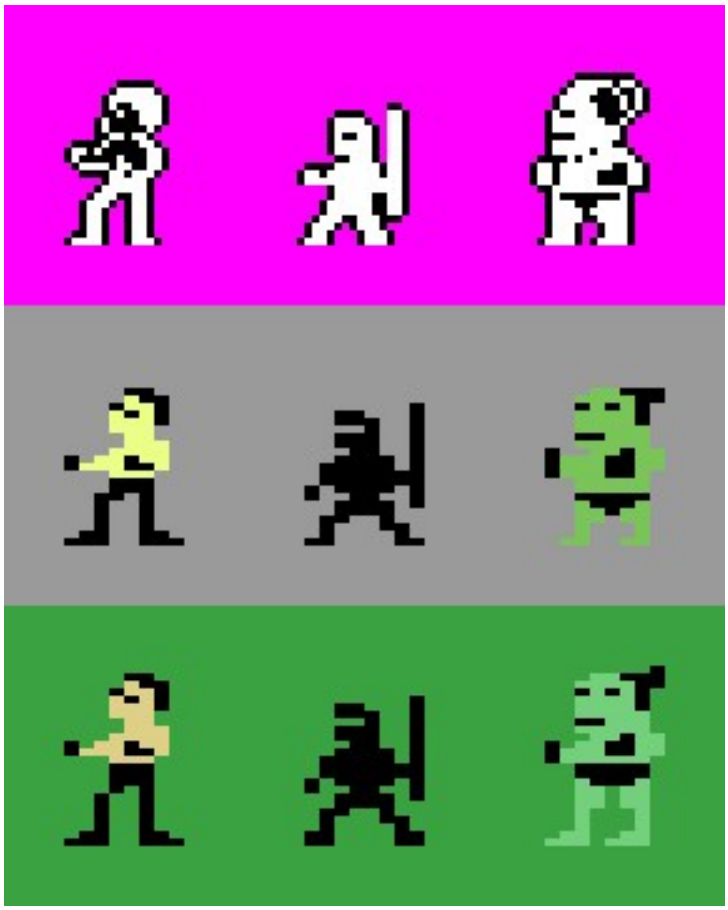


Figure 8. Bruce Lee characters Bruce, Ninja and Green Yamo (ZX Spectrum, Commodore 64 and MSX). The Spectrum row highlights the outlined sprite mask, which is not uniformly applied. The magenta color is not part of the game.

The movement and responsiveness are different from the Commodore 64 version, as the player and enemy character motion is locked to the tile resolution, meaning it is not possible to move the characters horizontally at pixel precision. This may relate to the challenges the developers faced when making smooth sprite-like visuals on the ZX Spectrum. As already noted, *Starquake* used rapid software sprites with horizontal movement variants pre-stored in memory (Figure 4). In *Bruce Lee*, the lock-step approach has likely helped the developers to use fewer frames of animation and shifted variants than would have been needed for freer positioning. As the sprites are masked, the mask definitions also consume memory. The software sprites move by four-pixel horizontal increments and have only two animation frames for running. As the vertical position computations require less effort, the characters can afford to fall smoothly.

All versions feature a relatively similar map, yet certain differences can be spotted, as the 40-tile wide rooms have been adapted to the MSX and ZX Spectrum versions as 32 tiles wide, condensing

the screen horizontally. This is the inverse of what was done with *Starquake's* Commodore 64 version. Comparing the room layouts between the two 32-character wide versions, the MSX and Spectrum, is the best indication that the development of the two had little or nothing to do with each other, as different layout choices have been made for the two maps. The MSX version maintains more of the look and feel of the C-64 original, despite the exceptional color choices, whereas the Spectrum graphics have been redrawn from scratch (see Figure 7).

Soundwise, *Bruce Lee* is minimalistic with only simple sound effects during gameplay. Bruce's footsteps, hits, kicks, traps and certain events trigger their own respective sounds but, all in all, the game is rather silent. This minimalism is particularly underlined in the Spectrum version, as the effects have been reduced to simple snaps and beeps. A little more effort is evident on the C-64, where the SID effects resemble physical sounds to at least some degree. These two versions play a short ”Oriental” tune at the beginning, whereas the MSX does not feature even that because of the missing title screen. In the same vein, a rather questionable portrayal, bordering on a racial stereotype, can be observed in the color choice of a yellow Bruce on the C-64 and MSX – interestingly, the Atari original used a more neutral skin tone (Figure 8; cf. Said 1994 [1978]).

Bruce Lee is a representative example of converting a Commodore 64 (or in this case, also Atari) title to platforms without matching features. Still, as the program is not especially complex, one might expect the conversions to at least play like the original. The varying results mostly suggest a lack of rigor in the development: design specifications might not have existed and a reasonably similar outcome was considered good enough to be sold. Yet, in this loose translation process, platform-related technical qualities also reveal themselves. Adapting the chamber width to a lower number of tiles, rather than forcing a narrower tile size, is a choice that readily emerges from the MSX and Spectrum display characteristics. Likewise, software sprites on the ZX Spectrum could have behaved more like their Commodore counterparts, but aligning them to the tiles is a practical Spectrum-specific technique.

Conclusion

When considering the pros and cons of comparative platform studies, it quickly became evident that the approach offers benefits as well as plenty of challenges that limit its applicability. To start off with the main positive finding, it is undeniable how the comparative approach offers – almost forces – a wider perspective when considering a platform and brings forth questions that would

otherwise go unnoticed. In the case of the 1980s' home computers, they were competing in a crowded, yet unsettled market, where technical features improved at a great speed, driven by and inspired by rivaling designs. Each machine came with its unique strengths and weaknesses, which facilitated certain kinds of content and, thus, converting or porting games from other platforms often required heavy, even impossible shoehorning when the hardware was not up to the task.

Comparing a number of different technical platforms makes their essential features more pronounced. For example, the importance of hardware sprites and scrolling stands out as other platforms may not have these features. This finding also invites examining in detail the different approaches for producing such visuals and how visual data is arranged in the computer memory. Especially the lack of hardware sprites brings forth questions about why a particular route was chosen for drawing routines and how they might have evolved during the course of the platform's existence. A ZX Spectrum ”sprite” is largely a culturally defined concept, which refers to the built-in features of other platforms. At the same time, the use of hardware sprites and sprite multiplexing on the Commodore 64 and MSX became more nuanced and complex over time – while technical features stayed the same, their use in games was never static. A comparative platform study should not list the presence or absence of features uncritically, but rather examine more broadly how the techniques for creating visuals and audio evolved over time.

Focusing on one platform alone would not have equally revealed the complex network of actors that were involved in the 1980s' game industry. Companies ranged from arcade giants to single-person cottage industries, with popular games flowing across borders. Ports, rereleases and region-specific versions call into question the concept of a monolithic ”game” as an object of study, as such an entity may not exist at all. The differences between versions could be likened to the changes in nuance that translations bring to written works. That said, a digital conversion can also have its own identity that exists without the original, at times diverging so far from its roots that the offshoot effectively becomes a different game altogether.

We need to keep in mind at all times that many games are the way they are because of other than hardware-related reasons. Tight deadlines, the lack of resources and skills, as well as poor documentation and tools are, in many cases, the real culprits behind low-quality versions. In the chosen examples, the platform characteristics more often provided straightforward routes for conversion rather than giving rise to ingenious solutions from the developers. Studying a certain game might reveal more about external issues than about the creative use of technology – most

titles were neither technically outstanding nor otherwise clever. Therefore, the selection of games that are analyzed largely dictates the results, too. Cherry-picking titles that illustrate one hardware feature or another particularly well, as often seen in platform studies, is understandable, yet does not represent the realities of the industry.

An interesting observation that a comparative study brought forth is how even the tiniest details affect the user experience. For instance, minute differences in game speed or responsiveness, as seen in *Bruce Lee*, lead to considerably different gameplay. Rendering the exact same graphics on different hardware can lead to crisp visuals or, in a less fortunate case, a flickering, slow and messy outcome. In the audio domain, the same tunes, when arranged for and played back on various sound chips, can appear either catchy or, at their worst, downright irritating. The same goes for sound effects, which were aplenty in every game we analyzed. As above, the differences cannot be attributed to one reason alone, but rather to a mixture of hardware features, author skills, tools and resources.

One notable challenge with the comparative approach is that there has to be *something to compare to* before any additional insight can be unearthed. An isolate platform with no contemporaries does not lend itself to such analysis. Likewise, a completely marginal machine with an accordingly modest software library does not facilitate direct comparisons across titles, even though other kinds of analyses are still doable. At the other end of the scale, there can also be an abundance of cross-platform titles and competitors, which leads to difficult choices in the face of unmanageable breadth; the colorful home computer market of the 1980s is a clear example of such.

From a practical perspective, studying more than one platform introduces considerable challenges and extra duties that one could otherwise avoid. First and foremost, learning about the history, technical details, software library and culture(s) of use of not one, but multiple platforms also multiplies the workload of the researcher, a problem that can best be alleviated by collaboration. In our case, too, the groundwork that involved finding working game versions and setting up both real hardware and emulators for the three machines required considerable effort. Even if platforms can be approached from different perspectives, certain ”hard” technical skills are still required, which effectively limits who can conduct such studies.

As our purpose here was to experiment with the concept of comparative platform studies, it is easy to point out multiple things that could have been done differently or expanded into a new direction. For practical reasons the focus was on commercial, notable titles instead of small indie releases that would highlight other aspects of the industry. Both the games and their underlying platforms represent three hotbeds of home computing and video games – the US, UK and Japan – and, thus, we did not break far away from the mold that platform studies has been criticized for.

As for future directions, there are numerous opportunities for similar comparisons between platforms: for example, the interplay between arcade originals and home versions offers one potential subject. The intertextuality of software already manifested itself in a number of ways, but as the topic is so vast and understudied, it could certainly be explored from several more perspectives. To conclude, based on our study, comparative *platform studies* does bring something new to the table, yet it is not an all-purpose tool for every case, as it introduces its own unique conceptual and practical challenges.

References

All links checked on November 2, 2026.

Websites

FRGCB Dude. n.d. ”FRGCB – Finnish Retro Game Comparison Blog”. https://frgcb.blogspot.com/

MIT Press. n.d. ”Platform Studies”. https://mitpress.mit.edu/series/platform-studies/

Sinclair Collection. n.d. ”Sinclair Collection”. https://www.sinclaircollection.site/

Magazines

Zzap!64, May 1987.

Literature

Adamson, Ian & Richard Kennedy. 1986. *Sinclair and the Sunrise Technology: The Deconstruction of a Myth*. Harmondsworth: Penguin Books.

Altice, Nathan. 2015. *I Am Error: The Nintendo Family Computer / Entertainment System Platform*. Cambridge, MA: MIT Press.

Apperley, Thomas & Jussi Parikka. 2018. ”Platform Studies’ Epistemic Threshold”. *Games and Culture* 13 (4), 349–69. https://doi.org/10.1177/1555412015616509

ASCII Corporation. 1984. *MSX Technical Data Book: Hardware/Software Specifications*. Atsugi-shi: Sony Corporation.

Avalon Software. 1985. *The MSX Red Book*. Pangbourne: Kuma Computers.

Bagnall, Brian. 2005. *On the Edge: The Spectacular Rise and Fall of Commodore.* Winnipeg: Variant Press.

Bogost, Ian & Nick Montfort. 2009. ”Platform Studies: Frequently Questioned Answers”. In Simon Penny (ed.) *Proceedings of the Digital Arts and Culture Conference 2009*. Irvine, CA: University of California. https://escholarship.org/uc/item/01r0k9br

Botz, Daniel. 2011. *Kunst, Code und Maschine – Die Ästhetik der Computer-Demoszene* [Art, Code and Machine – Aesthetic of the Computer Demoscene]. Bielefeld: Transcript Verlag.

Botz, Daniel. 2017. ”’If You Can Read This…’ – The Evolution of the Scroll Text Message Within the Demoscene”. *WiderScreen* 20 (1–2). http://widerscreen.fi/numerot/2017-1-2/if-you-can-read-this-the-evolution-of-the-scroll-text-message-within-the-demoscene/

Dillon, Roberto. 2015. *Ready: A Commodore 64 Retrospective*. Singapore: Springer. https://doi.org/10.1007/978-981-287-341-5

Fernández-Vara, Clara & Nick Montfort. 2013. *Videogame Editions for Play and Study*. Technical report TROPE–13–02. Cambridge, MA: MIT. http://hdl.handle.net/1721.1/87668

Gazzard, Alison. 2016. *Now the Chips Are Down: The BBC Micro*. Cambridge, MA: MIT Press.

Gere, Charlie. 2002. *Digital Culture*. London: Reaktion Books.

Grabarczyk, Paweł & Espen Aarseth. 2019. ”Port or Conversion? An Ontological Framework for Classifying Game Versions”. In *Proceedings of DiGRA 2019*. Digital Games Research Association DiGRA. https://doi.org/10.26503/dl.v2019i1.1108

Guillén, Claudio. 1993. *The Challenge of Comparative Literature.* Cambridge, MA & London: Harvard University Press.

Haddon, Leslie. 1988. ”The Home Computer: The Making of a Consumer Electronic”. *Science as Culture* 1 (2), 7–51. http://dx.doi.org/10.1080/09505438809526198

Huhtamo, Erkki. 2016. ”Amusement Arcade”. In Henry Lowood & Raiford Guins (eds.) *Debugging Game History: A Critical Lexicon*, 21–8. Cambridge, MA: MIT Press.

Hunt, Leon. 2003. *Kung Fu Cult Masters. From Bruce Lee to Crouching Tiger*. London and New York: Wallflower Press.

Ippolito, Jon. 2016. ”Emulation”. In Henry Lowood & Raiford Guins (eds.) *Debugging Game History: A Critical Lexicon*, 133–41. Cambridge, MA: MIT Press.

Jarratt, Steve (ed.). 2015. *Sinclair ZX Spectrum: a Visual Compendium.* Bath: Bitmap Books.

Jones, Steven E. & George K. Thiruvathukal. 2012. *Codename Revolution: The Nintendo Wii Platform*. Cambridge, MA: MIT Press.

Juul, Jesper. 2024. *Too Much Fun: The Five Lives of the Commodore 64 Computer.* Cambridge, MA: MIT Press.

Juul, Jesper & Laurel Carney. 2023. ”Would you like Games with that Computer? Revisiting early Game History & Culture with the Commodore 64”. In *Proceedings of DiGRA 2023*. Digital Games Research Association DiGRA. https://doi.org/10.26503/dl.v2023i1.1945

Kerr, Aphra. 2012. ”The UK and Irish Game Industries”. In Peter Zackariasson & Timothy L. Wilson (eds.) *The Video Game Industry: Formation, Present State, and Future*, 116–33. Abingdon: Routledge.

Kirkpatrick, Graeme. 2016. ”Making Games Normal: Computer Gaming Discourse in the 1980s”. *New Media & Society* 18 (8), 1439–54. https://doi.org/10.1177/1461444814558905

Lean, Thomas. 2014. ”’Inside a Day You Will Be Talking to It Like an Old Friend’: The Making and Remaking of Sinclair Personal Computing in 1980s Britain”. In Gerard Alberts & Ruth Oldenziel (eds.) *Hacking Europe: From Computer Cultures to Demoscenes*, 49–71. London: Springer. https://doi.org/10.1007/978-1-4471-5493-8_3

Leorke, Dale. 2012. ”Rebranding the Platform: The Limitations of ’Platform Studies’”. *Digital Culture & Education* 4 (3), 257–68. http://www.digitalcultureandeducation.com/cms/wp-content/uploads/2012/12/dce1073_leorke_2012.pdf

Lindsay, Christina. 2003. ”From the Shadows: Users as Designers, Producers, Marketers, Distributors, and Technical Support”. In Nelly Oudshoorn & Trevor Pinch (eds.) *How Users Matter: The Co-Construction of Users and Technology*, 29–50. Cambridge, MA: MIT Press.

Mailland, Julien & Kevin Driscoll. 2017. *Minitel: Welcome to the Internet.* Cambridge, MA: MIT Press.

Malazita, James. 2024. *Enacting Platforms: Feminist Technoscience and the Unreal Engine*. Cambridge, MA: MIT Press.

Marecki, Piotr, Yerzmyey & Robert Straka. 2020. *ZX Spectrum Demoscene*. Kraków: Jagiellonian University Press.

McAlpine, Kenneth B. 2018. *Bits and Pieces: A History of Chiptunes.* New York: Oxford University Press.

Montfort, Nick & Ian Bogost. 2009. *Racing the Beam: The Atari Video Computer System*. Cambridge, MA: MIT Press.

Morris, Meaghan. 2004. ”Transnational Imagination in Action Cinema: Hong Kong and the Making of a Global Popular Culture”. *Inter-Asia Cultural Studies* 5 (2), 181–99. https://doi.org/10.1080/1464937042000236702

Newman, James. 2012. *Best Before: Videogames, Supersession and Obsolescence*. Abingdon: Routledge.

Newman, James. 2017. ”Driving the SID Chip: Assembly Language, Composition, and Sound Design for the C64”. *G*|*A*|*M*|*E* 1 (6).

Oliveira, Thaiane, Emmanoel Ferreira, Louise Carvalho & Andre Boechat. 2014. ”Tribute and Resistance: Participation and Affective Engagement in Brazilian Fangame Makers and Modders’ Subcultures”. *G*|*A*|*M*|*E* 3/2014. https://www.gamejournal.it/3_oliveira_et-al/

Penix-Tadsen, Phillip. 2024. ”Pirating Platform Studies: The Historical Impact of Latin American Clone Consoles, 1973–1994”. *Gamevironments* 20, 35–87. https://doi.org/10.48783/gameviron.v20i20.248

Remak, Henry. 1961. ”Comparative Literature, Its Definition and Function”. In Newton P. Stallknecht & Horst Frenz (eds.), *Comparative Literature: Method and Perspective*, 3. Carbondale, IL: Southern Illinois University Press.

Reunanen, Markku, Guus Hoeberechts & Tytti Viljanen. 2024. ”’A Step into the Computer Era’ – A Comparative Study on Early Home Computing in the United Kingdom, the Netherlands, and Finland”. *WiderScreen Ajankohtaista* 26.9.2024. http://widerscreen.fi/numerot/ajankohtaista/a-step-into-the-computer-era–a-comparative-study-on-early-home-computing-in-the-united-kingdom-the-netherlands-and-finland/

Reunanen, Markku, Petri Saarikoski & Tapani Joelsson. 2025. ”’Kaverien kanssa ruvettiin harrastamaan ATK:ta’: Tietokoneiden tulo 1980-luvun suomalaisiin koteihin” [’Me and Friends Got into Computing’: The Arrival of Computers to Finnish Homes in the 1980s]. *Kulttuurintutkimus* 42 (1), 19–38.

Reunanen, Markku, Tapani Joelsson & Petri Saarikoski. 2024. ”Commodoren varjossa: 1980-luvun tietokonepelaamista marginaalikoneilla” [In Commodore’s Shadow: Playing on 1980s’ Marginal Home Computers]. In Jonne Arjoranta, Pauliina Baltzar, Jani Kinnunen, Lilli Sihvonen, Heidi Rautalahti & Tanja Välisalo (eds.) *Pelitutkimuksen vuosikirja 2024*, 2–25. Tampere: Finnish Society for Game Research.

Reunanen, Markku, Tero Heikkinen & Anders Carlsson. 2018. ”PETSCII – A Character Set and a Creative Platform”. *Replay: The Polish Journal of Game Studies* 5 (1), 27–47. https://doi.org/10.18778/2391-8551.05.02

Saarikoski, Petri & Markku Reunanen. 2014. ”Great Northern Machine Wars: Rivalry Between User Groups in Finland”. *IEEE Annals of the History of Computing* 36 (2), 16–26. https://doi.org/10.1109/MAHC.2014.20

Said, Edward W. 1994 [1978]. *Orientalism*. New York: Vintage Books.

Salter, Anastasia & John Murray. 2014. *Flash: Building the Interactive Web.* Cambridge, MA: MIT Press.

Smith, Chris. 2010. *The ZX Spectrum ULA: How to Design a Microcomputer*. Dinas Powys: ZX Design and Media.

Stachniak, Zbigniew. 2015. ”Red Clones: The Soviet Computer Hobby Movement of the 1980s”. *IEEE Annals of the History of Computing* 37(1), 12–23. https://doi.org/10.1109/MAHC.2015.11

S.T.A.R. 2012. *The Legend of Konami: The Book*. N.p.: Matra Computer Automations.

Suzuki, Mana. 2016. ”Inquiry into the Relation Between Japanese Word Processors and the MSX Personal Computer Series”. *Philosophy and History of Science Studies* 10, 70–7. https://doi.org/10.14989/210108

Švelch, Jaroslav. 2018. *Gaming the Iron Curtain: How Teenagers and Amateurs in Communist Czechoslovakia Claimed the Medium of Computer Games*. Cambridge, MA: MIT Press.

Therrien, Carl. 2012. ”Video Games Caught up in History: Accessibility, Teleological Distortion, and Other Methodological Issues”. In Mark J. P. Wolf (ed.) *Before the Crash: Early Video Game History*, 9–29. Detroit, MI: Wayne State University Press.

Tomczyk, Michael S. 1984. *The Home Computer Wars: An Insider’s Account of Commodore and Jack Tramiel*. Greensboro, NC: Compute! Publications.

Veraart, Frank. 2008. ”De Domesticatie van de Computer in Nederland 1975–1990” [The Domestication of the Computer in the Netherlands 1975–1990]. *Studium* 1 (2), 145–64. https://doi.org/10.18352/studium.1460

Vuorinen, Jukka. 2007. ”Ethical Codes in the Digital World: Comparisons of the Proprietary, the Open/Free and the Cracker System”. *Ethics and Information Technology* 9 (1), 27–38. https://doi.org/10.1007/s10676-006-9130-2

Wade, Alex. 2016. *Playback – A Genealogy of 1980s British Videogames*. London: Bloomsbury.

Wasiak, Patryk. 2012. ”’Illegal Guys’. A History of Digital Subcultures in Europe during the 1980s”. *Zeithistorische Forschungen/Studies in Contemporary History* 9 (2), 257–76. https://doi.org/10.14765/zzf.dok-1595

Woods, Kam & Geoffrey Brown. 2008. ”Virtualization for Preservation of Executable Art”. In *Proceedings of the 2008 DOCAM Summit and Symposium*. Montreal: DOCAM.

Yamashita, Ryozo. 2020. ”History of Personal Computers in Japan”. *International Journal of Parallel, Emergent and Distributed Systems* 35 (2), 143–69. https://doi.org/10.1080/17445760.2019.1587435