Reunanen, Markku; Heikkinen, Tero; Carlsson, Anders

PETSCII – A Character Set and a Creative Platform

Published in:
Replay

Published: 22/11/2019

Document Version
Publisher's PDF, also known as Version of record

Please cite the original version:
Reunanen, M., Heikkinen, T., & Carlsson, A. (2019). PETSCII – A Character Set and a Creative Platform. *Replay*, *5*(1), 27–47. https://czasopisma.uni.lodz.pl/Replay/article/view/5930

Replay 1(5) 2018
http://dx.doi.org/10.18778/2391-8551.05.02

Markku Reunanen
University of Turku and Aalto University

Tero Heikkinen
University of the Arts Helsinki

Anders Carlsson

PETSCII – A Character Set and a Creative Platform

Introduction

Throughout the 1980s Commodore International was arguably the king of the home computer market, thanks to its successful VIC-20, Commodore 64 and the Amiga line of computers (e.g. Bagnall, 2005). The wildly popular Commodore 64, in particular, became an iconic machine of the home computer boom of the early 1980s. One factor contributing to its recognizable aesthetic was its PETSCII character set which the 1980s enthusiasts still remember from the blue startup screen stating that there were "38911 BASIC BYTES FREE". Even if the C-64 was capable of various pixel graphic modes, its text mode, too, found its uses and has recently become popular again, as a number of visual artists have started experimenting with the format and pushing its boundaries over the last few years.

Before we begin, it is necessary define what "PETSCII" means, as the term has different meanings depending on the context. Technically speaking, it refers to Commodore's *PET Standard Code of Information Interchange*, a character encoding used on Commodore's 8-bit computers that partially overlaps with the widely used ASCII encoding. In practice, however, the term is commonly used to denote the built-in character set and font of the Commodore 64, which is also the focus of this article. As can be seen in Figure 1, there are minor differences in the character set between machines and their revisions: for example, European C-64s featured a pound sign instead of a backslash. Most PETSCII graphics are marked by the technical aspects of the C-64, such as its fixed

palette of 16 colors, a text-grid of 40 by 25 characters, and a resolution of 320 by 200 pixels. See Carlsson's (2017) overview of text graphics for further definitions.

Figure 1. The PET and C-64 fonts (the default and "shifted" set that has both upper and lowercase letters). Shown in screen code, not in PETSCII order here.

In spite of its longevity, continuing popularity and importance for the history of home computing, there are few academic takes on the subject so far – computer-made text art in general is a little studied topic, even if there are known examples dating back at least to the 1960s (Franke, 1971). The largest PETSCII-related publication so far is the *Software Studies* series book *10 PRINT CHR$(205.5 + RND(1)); : GOTO 10* (Montfort et al., 2012) where a short BASIC language program is analyzed in depth, leading to discussions on wider topics, such as randomness in software and grid-based art. Raquel Meyers provides a personal, artistic manifesto on her paper *Keys of Fury*, where she brings together multiple strands of text art (Meyers, 2017; see also Paul, 2013). Likewise, Tommi Musturi's (2017) reflections on his own demoscene works shed light on artists' needs and working methods.

The main research question of this article can be stated as follows: *How is the PETSCII character set used in creative works?* As a secondary aim we also seek answers to how tools have enabled – or hindered – such endeavors. Our primary research material consists of PETSII-based tools, games and still images. In addition, we present and reflect upon our own editor project that has provided us with practical grassroots perspective on how it is to actually work with the character set.

Bits of History

Text art has a long and rich history with Christian, Muslim and Jewish examples that date back at least 2500 years (Carlsson & Miller, 2012). PETSCII's most recent ancestors, however, are text graphics made with computers, typewriters, teletypes and

other digital and electronic communication technologies from the 19th and 20th century (Carlsson & Miller, 2012). However, as PETSCII contains a rather extensive set of graphic symbols and allows the use of colors, it often looks significantly different from traditional monochrome ASCII art that focuses on alphanumeric signs and other typographical symbols. In this respect, PETSCII has more in common with ANSI, teletext and other colorful blocky text graphics characterized by the use of geometric shapes (see Hardagon, 2011; Moe & Van den Bulck, 2016).

As such, this style can be called *text mosaic* (Carlsson, 2017) because it taps into ancient arts and crafts such as mosaics, weaving and, as noted by Danet (2003), quilting. In short: PETSCII places geometric shapes on a fixed grid. Grids in art are universal (e.g. Higgins, 2009; Montfort et al., 2012), but often hidden or used in organic ways, as in perspective methods or proportional systems. Instead, the grid was hardwired in most 1980's computers: all the symbols of the character set had the same size, and were positioned on the screen according to a fixed grid. Although it is possible to transgress these grids, especially with modern tools, the grid remains an fundamental part of the aesthetics of PETSCII and many other forms of text art.

Figure 2. Commodore PET PETSCII graphics. *Dog Star Adventure* (1980) by David Malmberg.

Early home computers often relied on a character display mode, which, compared to bitmap graphics, requires less memory and less processing power. The PETSCII character set, designed by Chuck Peddle and Leonard Tramiel (Bagnall, 2005, 54–55),

first appeared on the Commodore PET from 1977 (see Figure 2 for an early example). The later home-oriented VIC-20 with color graphics also had PETSCII as the default set, but the most significant variant was created for the Commodore 64. On the C-64, the lines were made thicker, assumably to make text more readable on a television screen; on the VIC-20 the thickness was less of an issue owing to its lower resolution, and the PETs came with their own display. The later models, the Commodore 128, Plus/4 and C-16, have a similar thick font with only minor adjustments.

During the 8-bit consumer home computing era it was typical to have a BASIC language interpreter and the operating system on ROM chips. The Commodore home computer line contained the PETSCII character set and a BASIC interpreter from which the set could be accessed. This made both the BASIC and the visuals a de facto standard. Notably, the graphics could be mocked up using the full-screen editor without having to program anything. Many early Commodore 64 games used the character set in versatile ways; a few examples can be seen in Figure 3. As the games industry became more professional, commercial C-64 game visuals moved towards smooth machine code bitmap- and sprite-based graphics. PETSCII and character visuals began to signify old-fashioned, hobbyist type-in software (cf. Saarikoski, Suominen & Reunanen, 2017).

Figure 3. Early PETSCII-based Commodore 64 games. *Back to Nature* (1982) by Commodore, *Alien* (1984) by Softgold, and *Murder* (1983) by Rabbit Software.

While games moved away from text graphics, character-based screens continued their life on, for example, bulletin board systems (BBS). As bulletin boards were accessed with low-speed modems and different computers, ASCII most often provided both a common standard and a memory-efficient solution. In addition, C-64 specific boards could use PETSCII graphics. Artists made menus, logotypes and even animations, mostly with the lowercase PETSCII character set. As bulletin boards lost most of their users to the Internet in the late 1990s, PETSCII graphics were less popular over the next two decades, save a few enthusiasts, such as Poison (Radek Stypczynski).

In the early 2010s, the Californian artist Max Capacity started to work with PETSCII without using Commodore computers. He used the character set with modern tools, which enabled him to use higher resolutions and more colors, and to make longer animations than the 64 kilobytes of the C-64 could handle. He mostly used the standard grid, which made his PETSCII works fit within the traditional PETSCII aesthetics. Max Capacity published his works online on Flickr and Tumblr, and gained quite a following for his works. More recently, in 2016, an artist called Ailadi published daily PETSCII graphics, that sparked interest around the Web.

Max Capacity and Ailadi are part of a new wave of PETSCII graphics that is not necessarily connected to the Commodore 64. Meanwhile, the C-64-dependent PETSCII graphics have been heavily developed within the demoscene since a new revival that began around 2013 and raised the bar for such works. The Plain PETSCII Competition held online on the *Commodore 64 Scene Database* website in the fall of 2013 created notable enthusiasm towards the format, and also led to the creation of new authoring tools. The time and effort invested in such activities illustrate the role of enthusiasts in keeping a platform alive way past its prime (see Lindsay, 2003).

PETSCII has also been used for generative experiments with less focus on realism. The book title *10 PRINT CHR$(205.5 + RND(1)); : GOTO 10*, already mentioned earlier, is a program that creates a maze-like PETSCII pattern by randomly printing either / or \. This *oneliner* also inspired the New York Public Library to generate book covers for books that had no cover (Giraldo Arteaga, 2014). Similar approaches can be found in the C-64 demoscene; for example, Matthew "4mat" Simmonds made small generative PETSCII programs that reacted to music for the *dubCRT* C-64 cartridge.

A Tool Perspective

It is possible to doodle text graphics directly on the screen using the BASIC interpreter's standard editor; all textual characters and graphic symbols are accessible on the keyboard through key combinations. The colors of the characters can also be changed the same way. Most Commodore keyboards had the symbols displayed on the keys, facilitating editing. The main issue with this rudimentary approach is that there is no easy way to save or load the works afterwards, unless you have a separate "freezer" cartridge at hand. A less direct way is to build the image out of character strings and use BASIC's PRINT statements to output them on the screen – or a printer, for that matter (see Heikkinen & Reunanen, 2017).

For more extensive work, a text graphics editor is a necessity. For this study we analyzed the functionality of 14 different, publicly available C-64 PETSCII editors, some of which run on actual hardware and some on modern platforms (Table 1). Roughly speaking, native editors represent the older end of the spectrum, some dating back as far as the 1980s, whereas newer options tend to be cross-development tools created during the last five years. At times it was impossible to reliably establish the release date, as there was no mention of it in the program or the file archive. Moreover, as there are several versions of the editors around, they could be considered as continuums rather than well-defined single artifacts.

Native C-64 Editors	Cross-Development Tools
Digital Paint	*Another PETSCII Editor* (Windows)
Kaleidoscope	*C-64 ASCII Art Editor* (Windows)
PETSCII Editor	*CBM .prg Studio* (Windows)
PETSCII Paint	*EDSCII* (Linux, Windows, Mac)
TBoard-Painter Pro	*Online PETSCII Editor* (online)
Tyronpaint	*PET Shop Pro* (online)
	PETSCII (Linux, Windows, Mac)
	Playscii (Linux, Windows, Mac)

Table 1. A list of the tested PETSCII editors.

Digital Paint (Figure 4, probably the late 1980s), *Kaleidoscope* (1989) and *Tyronpaint* (1996) are three examples of early editors. All of them feature basic typing and editing functionality with some curious extras: *Kaleidoscope*, for example, can automatically change the character color ("rainbow mode") and *Tyronpaint* lets the user draw rough 80x50 pseudographics using the ¼ block characters of PETSCII. All the old editors use a black background by default, which hints at their potential use for BBS graphics.

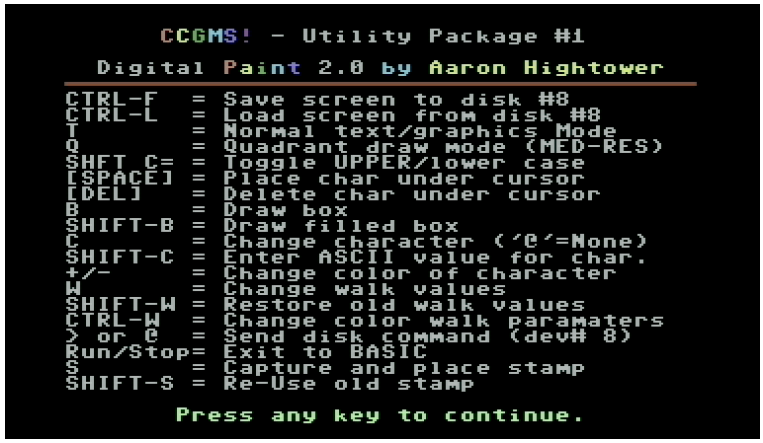

Figure 4. *Digital Paint* help screen.

Not all the native tools are old. For instance, 0xDB's simple *PETSCII Paint* is from 2011. The most developed editor running on an actual C-64, *PETSCII Editor* by fieserWolf, was last updated in 2017. The first released version of the program dates back to 2004, after which there have been multiple improved releases. Compared to its predecessors, *PETSCII Editor* features advanced editing functionality, such as multiple workspaces, recoloring and copy/paste for regions. Like in all the native editors, the principal way of working is typing, as there is no mouse support – there were mice available for the C-64, but they never became commonplace.

Cross-development tools running on today's platforms have much more resources at their disposal: a mouse, screen space for a graphical user interface, processing power, memory, Internet connection and rapid storage. Commonplace features, such as large undo/redo buffers and animation, would be impossible or at least tedious to implement on a C-64.

In spite of the undeniable technical and usability reasons for using cross-platform tools, there are other, cultural factors that can hinder their use: questions of authenticity are good enough reasons for some users to keep working on real hardware. For example, reading Raquel Meyers' (2017) artistic reflection gives an impression of almost spiritual commitment to the 1980s' physical keyboard. Apart from those issues, the possibility to immediately see the outcome as it looks like on an actual C-64 is a concrete plus, as well as having the PETSCII characters visible on the keyboard.

PETSCII Editor v2.0, also known as *Online PETSCII Editor*, by Krisztián Tóth is a full-featured text graphics editor, which runs in a browser window (http://petscii.

krissz.hu/, Figure 5). Using the Web as the application platform illustrates, among other things, how the latest technologies are quickly put into use when creating tools for retro computers. Apart from a large set of editing-related features, *PETSCII Editor* also lets the user edit the character set, unlike most other tools. In spite of all of its functionality, *PETSCII Editor* is, at its heart, still a purist tool that adheres to the technical capabilities of the C-64. Therefore, it is also possible to export the end result as an executable file (.prg) that runs on original hardware.

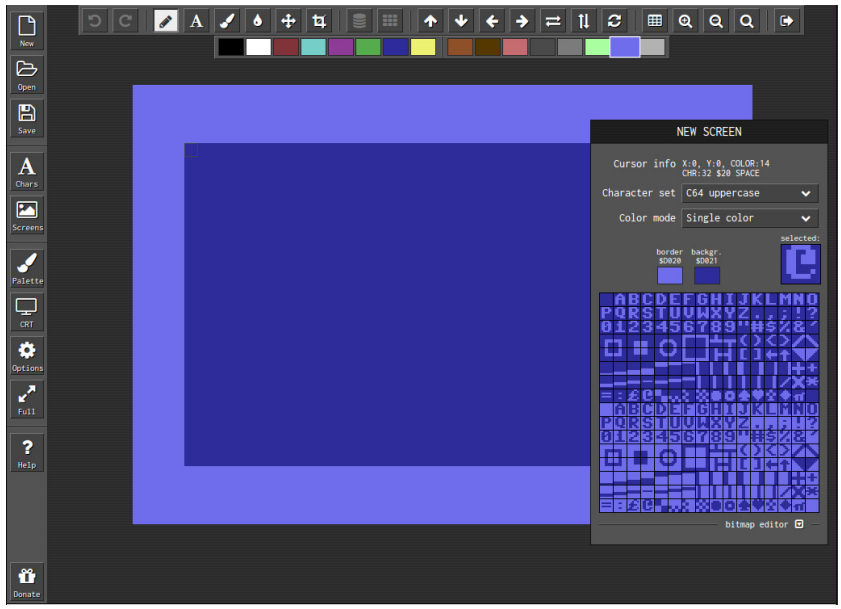

Figure 5. *PETSCII Editor v2.0.*

Playscii (2014) and its predecessor, *EDSCII* (2013), both by JP LeBreton, take a more relaxed approach to text art. Instead of sticking to the capabilities of existing machines, LeBreton's editors let the user do "impossible" things, such as freely setting a separate color for both the foreground and background of a character. It is also possible to change the color palette, which none of the 8-bit Commodores could do. Such an approach, on the one hand, eases the creative process and, on the other hand, turns "PETSCII" or any other available character set into a stylistic surface. *Playscii* is full of features familiar from common paint programs and, as a rarity, also lets the artist use layers as in photo editing software.

In fall 2013 there were less tools available, so two of us, Markku Reunanen and Tero Heikkinen, set out to implement a simple editor for our own use in the *Processing* environment (see Reas & Fry, 2014). The project, simply called *PETSCII* (http://www.kameli.net/marq/?page_id=2717), rapidly grew into a useful piece of software, and was released to the public shortly thereafter. Four years of development, constant personal use and reflection are the basis for the following discussion, where we highlight some of the design decisions and reasons behind them.

The character selector is at the very core of mouse-based editors. One of our first inventions was to reorganize the otherwise unintuitive character map – somewhat in the same lines as they are on the 8-bit Commodores' keyboards – to bring forward closely connected shapes. For example, line characters and different kinds of corner pieces should be close to each other. As can be seen in Figure 6, a simple reordering has made the graphical symbol set noticeably easier to approach. A similar remapping has become a standard feature in other recent tools as well, as can be seen in Figure 5.

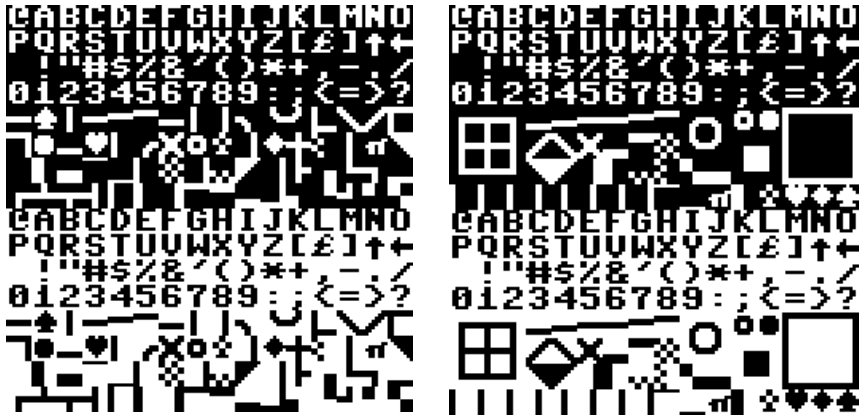

Figure 6. The default (screen code order) and manually reordered C-64 character set.

When looking at the geometric symbols it soon becomes evident how systematic they are, save a few exceptions. At times the necessary characters can be found on the inverted side of the set, as in the case of the triangles. This insight led to the realization that larger areas, even the whole screen, can often be mirrored or rotated without disturbing the shapes. The observation also encouraged us to develop a flexible selection tool, so that rectangular areas can be picked up quickly

and used as a brush immediately afterwards. Rotation and flipping tools that utilize the symmetry have proven to be powerful in daily use. Visually interrelated symbols can also belong to sets, which can be quickly skimmed through with a keystroke.

There are two inherent layers to a PETSCII image: the characters and the colors. It is often beneficial to separate the two when creating an image, for instance, by drawing the shapes first and coloring the graphics only in the end. Both *PETSCII editors*, among others, support this kind of workflow. Recoloring was considered so important that we implemented multiple features for it in our own *PETSCII* editor: a selection can be used as a "magic lens" that quickly recolors a large area, and individual colors can be replaced by others in order to try out different color schemes for an image.

In the big picture, metaphors borrowed from pixel-based paint programs – in our case, the venerable *Deluxe Paint* (see Maher, 2012) – are a useful starting point, but cannot be used as is. Instead of placing pixels, the main challenge is to select suitable characters and test alternatives as efficiently as possible. The simplicity of the late 1970s' and early 1980s' home computers makes today's common metaphors, such as translucent layers, less useful than in their schoolbook use, photo editing. Text graphics can be scaled up or down in marginal cases only. Likewise, drawing curved shapes, such as ellipses and arcs, has little use for a text artist. In contrast, PETSCII provides for simple line graphics and boxes, which can be supported by their respective tools.

For our research, the tool-creation process acted as a springboard for our questions and thoughts on text art and PETSCII. Devising images with an incomplete tool, adding new functions as deemed necessary, we were in a position to experience the tool at its various stages of maturity. Working from a simple editor with few options towards a fully-fledged package, we could experience "multiple" tools instead of just the final outcome. This gave us sensitivity for the idea that PETSCII works are made with different tools and artistic motives, setting the tone for our research. We had to consider a wider perspective: instead of only analyzing extant works, as developers we needed to ask what kind of approaches and tools facilitate the creation of text art. As Lev Manovich (2001, p. 118) states, the use of software is a two-way process: a tool does not only facilitate our work, but also affects how we perceive ourselves, others and the world.

Thus, our overall research incorporates elements of reflective research (Schön, 1991), embedded into our practice as researchers, but also draws from our experience as hobbyists, artists and computer aficionados. Such a research approach cannot be fully explicated as rational steps, as it also involves much tacit knowledge (Polanyi,

1966). To a certain extent, the above describes some of the reasoning that took place during the development, but it is possible to see the PETSCII tool we created also as an artifact that ties together our understanding of the topic.

Pictures out of Symbols

Most PETSCII graphics are made for the Commodore 64 and its default font, resolution, palette, memory and other characteristics, and this section focuses on such C-64 PETSCII. We have chosen 30 works from five recent PETSCII competitions: Plain PETSCII Graphics Competition 2013, Zoo 2013, Zoo 2015, X 2016, and Dir Art Compo 2017. In order to build a fair overview of current trends, the works were selected with even sampling to represent both the highest and the lowest ranking competition ("compo") entries. As there are works released outside competitions, too, we balanced the set with five additional pieces found on the *Commodore 64 Scene Database* (CSDb, http://csdb.dk/). See the Appendix for a full list with authors and publication years.

One notable characteristic of our selection is that it originates from the demoscene community. The framing might marginally skew the results, but on the other hand it is true that the scene is, at the moment, clearly the most active venue where PETSCII is created and published. The community is mostly male and has its own particular values colored by competition, meritocracy and high appreciation of technical skills (Silvast & Reunanen, 2014; cf. Lindsay, 2003). Geographically thinking, PETSCII graphics started as an almost worldwide phenomenon, but the second wave that we are experiencing now is markedly Eurocentric.

Before we move on to the strategies and techniques seen in the works, it is necessary to split contemporary PETSCII art into finer subcategories, as not all of them come with similar properties and limitations. As per our own observations and the CSDb taxonomy we can observe at least the following kinds:

- *Standard C-64 text mode* PETSCII utilizing the 40x25 character resolution, uppercase character set, one background color and 16 foreground colors.
- *Directory (dir) art* – 16 character wide images that fit into a floppy directory listing. Colors and inverted characters are not available by default without hacking, but the pictures can be higher than the standard 25 rows.
- *BBS graphics* – images with a black background that commonly rely on the shifted character set, which has both lowercase and uppercase characters, but notably less geometrical shapes (Figure 1).

- *Extended PETSCII-based graphics* with a different font, text mode or flicker colors that go beyond the capabilities of the standard text mode. Omitted here.
- *Fakescii* – images that look like PETSCII but do not conform to the capabilities of the original platforms, such as palette, memory footprint, color limitations or resolution. Omitted here.

In addition to static images, which represent the majority of all the works, some of the above categories also include animations that add a temporal dimension to the content. Following the text graphics model (Carlsson, 2017) we can describe the selected body of works using the following criteria:

- *Symbols:* The PETSCII-standard contains Latin letters, typographic and mathematical symbols, semi-graphic symbols, and non-printing characters.
- *Encoding:* The symbols are encoded in PETSCII, based on the 1963 ASCII standard and not the more common 1967 ASCII standard.
- *Production:* The works were made with different platforms, software, and techniques but mostly in the context of the C-64 and its demoscene.
- *Design:* All of the selected PETSCII graphics use the original fixed-width PETSCII font of the C-64, positioned in the default grid and in the original resolution.
- *Format:* The works are executable C-64 programs, except for directory art, which are file listings.
- *Display:* The works can be displayed either on a CRT screen with artifacts, such as pixel bleeding, or on modern screens without CRT artifacts.
- *Perception:* Most of the works are entries for C-64 demoscene competitions, which was the primary context of perception.

One of the most defining factors for an image is its *background color*. "Background" here is a rather technical term that refers to the color that all the character positions share. Visually thinking the color might not appear at all as background, because inverted characters, in effect, make it the "foreground" color. Therefore, it is more fruitful to think in terms of a *common color* – something that is available at every location on the grid. The *border color*, visible outside the character area, is the other freely selectable color affecting the overall mood of the image. The border either

creates a rectangular frame around the actual graphics or, alternatively, lets the image extend all the way to the edges of the physical display.

There are a few different strategies that artists have employed when dealing with the common color. First of all, black is the most usual choice for at least two reasons: it is required for cartoonish outlines and notably darker than any other color, providing high contrast if needed. Not all images are like that, however. Some artists set the common color to a neutral one and then place lighter and darker shades next to it to simulate light and shadow, much like in classical drawing, where you start with a midtone paper, and use chalk and charcoal for light and dark regions.

In the selected works we have identified three major visual styles: line-drawing, text mosaic, and pixel style. *Line drawings* use box-drawing characters and other line-based symbols to create an aesthetic close to ASCII art, as described by Carlsson (2017). The technique is frequently used for menu designs on bulletin boards and games, as well as in directory art, but is less popular for standalone PETSCII graphics.

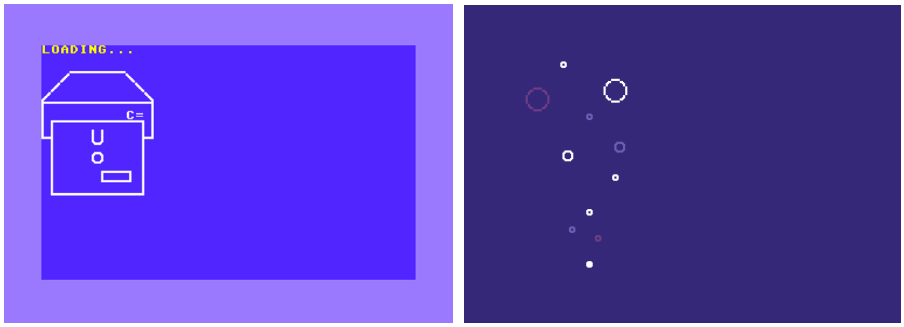

Figure 7. Line drawing examples. *Submerged* (2015) by Terwiz, and a single frame grabbed from the *Diskplosion* (2017) dir art animation by Logiker. See also Figure 2.

PETSCII line drawings were popular in the 1980s and are today often used to evoke nostalgic memories of early C-64 BASIC programs. As such, they tend to use the original blue background color, as shown in Figure 7. Line drawings are often seen as a naïve art form in the context of the competitive demoscene, and in our data such works were mostly seen in the directory art competition.

Text mosaics form shapes out of blocks rather than lines. The blocks are usually geometric shapes like squares, triangles or circles, and ideally each character is clearly distinguishable from other nearby characters. Occasionally, repetitive patterns of

characters or character combinations are seen as interesting in themselves, and form the main attraction of a more abstract picture.

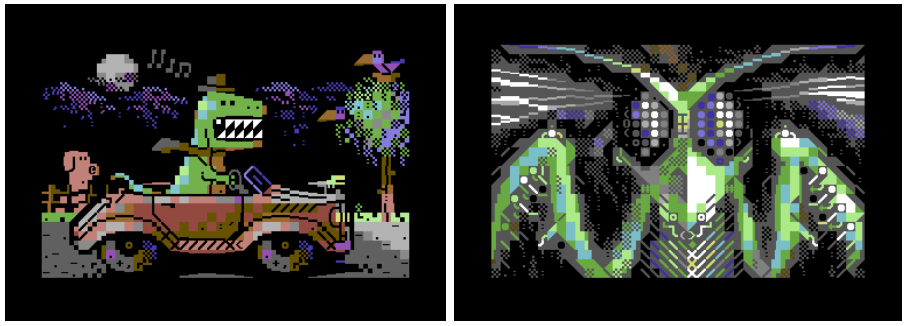

Figure 8. Text mosaic examples. *Gary* (2013) by Mermaid, and *Mantis Compo Version* (2013) by Slayer Grafix.

Pixel style is a technique to create details on sub-character level. The artist can use block elements such as ▄ and ▬ to achieve a crude pixel art style, but a key element of pixel style PETSCII is to find bridges between text characters. Consider, for example, how the small green alien in Redcrab's *BOSSE* (Figure 9), has a right leg made of a 7, and a body and mouth created with an inverted apostrophe. Tricks like these can make it difficult for the viewer to see where one character ends and the next begins.

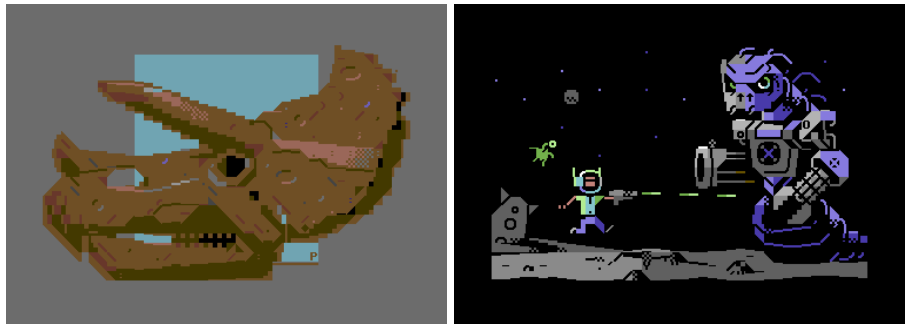

Figure 9. Pixel style examples. *Triceratopetscii* (2016) by ptoing and *BOSSE* (2016) by Redcrab.

The three categories of techniques – line drawings, text mosaic and pixel style – each come with conditions that encourage the artist to work towards a certain aesthetic. Line drawings promote what is often considered as a naïve and minimalistic

art form, text mosaics encourage blocky and colorful textural patterns, and pixel style typically pushes the artist to work with intricate details.

A PETSCII piece does not need to rely on one technique only. For example, in *Gary* (Figure 8) Mermaid uses text mosaic to form the body of the car while using pixel style and line drawings to accentuate details and make outlines for the car. The background has been fleshed out from very few characters. This has been achieved by seeing the abstraction potential in the characters and neighbour-combinations. Double quotes are used for the eyes, a triangle for the birds' beaks and an @ for the pig's snout.

This combination of techniques is vital for the recent trend of photorealism in PETSCII. The artist tries to mimic photographic realism with a minute attention to colors, shapes and ornamentation. The foundation of a photorealistic PETSCII piece is usually made with a sort of text mosaic using the varied densities of the characters, a technique that Xu, Zhang and Wong (2010) call tone-based. With this technique you can make a gradient from the denser characters (▄ X, #) to less dense characters such as . or ,. The same has previously been done in, for instance, block ASCII, typewriter art and teletype art to make monochrome graphics (Carlsson & Miller, 2012), but PETSCII's wide repertoire of semi-graphic symbols expands the possibilities and enables the artist to add pixel-style details. Colors also bring new potential for the artist to explore.

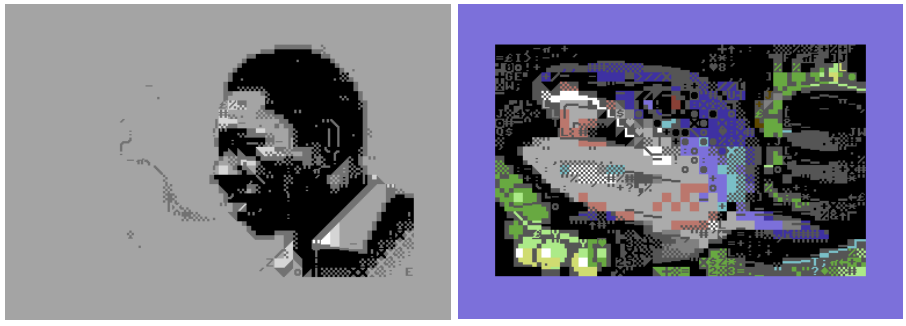

Figure 10. Figurative examples. *Coltrane* (2015) by Electric and *Hunter in the Shallows* (2013) by Hammerfist.

The works can hardly reach photorealism, given the resolution and palette at hand (Figure 10). When moving away from the capabilities of the original machines to "fake-scii" with its higher resolutions and more colors, PETSCII's potential for photorealism obviously increases. In an extreme case the symbols disappear altogether and converge with normal pixel graphics. Although all of the above techniques utilize cartoon-like

exaggeration and simplification, there are even more concrete examples in images where large color areas are combined with black outlines, such as a rendition of Asterix, Pikachu or Black Dynamite – all cartoons to begin with.

Low-resolution text art does not easily lend itself to complex depth effects. Most of the figurative PETSCII works have one or two foreground objects over an abstract or simple background. Overlapping is a common depth cue between multiple elements. Again, Mermaid's *Gary* serves an example of further complexity, with multiple overlapping elements in the same picture. The limitations of PETSCII have not prevented authors from attempting perspective effects, such as foreshortening (Figure 10), a hint of one-point perspective (Figure 7), or the slightly oblique cassette tape in *We Are All Ejected* (Figure 11).

Some approaches can be considered as tricks, applying a bitmap mindset for creating character graphics. Anti-aliasing is generally a bitmap technique where a middle color is used to soften the aliasing effect between the boundary of two colors. Anti-aliasing cannot be as universally applied to a PETSCII picture, but some artists have been able to add anti-aliasing style touches by carefully choosing neighboring characters and their colors (Figures 8 and 10).

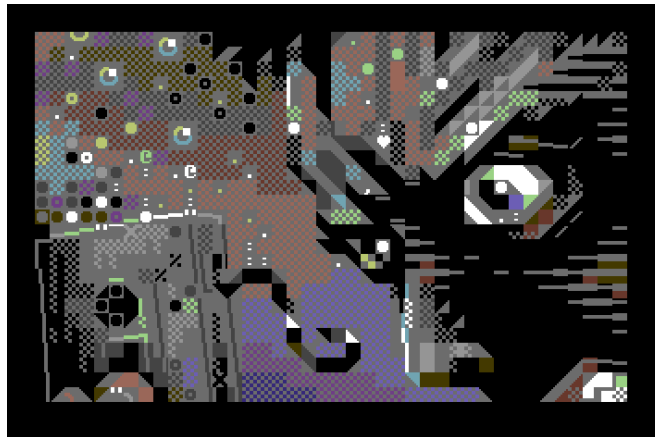

Figure 11. *We Are All Ejected* (2016) by Archmage. Raster colors are overlaid on a grey background.

Another well-known trick, used on a number of other platforms as well, is *rasterizing*. In bitmap images, dither patterns may be used for creating darker and lighter tones or even seemingly new colors. Although many PETSCII images use the varying character density for shadowing and tones, it is difficult to produce any illusion of new colors

with characters. The most straightforward way of rastering is to use the checkerboard characters in the PETSCII set (Figure 11). As can be seen, the thick 4 by 4 checkerboard of the C-64 is barely adequate for the purpose, whereas the original 8 by 8 version would work noticeably better, in particular on blurry TV screens (see Figure 1).

When considering the themes seen on the analyzed works, it is evident how PETSCII art is affected by external influences, popular culture in particular. *Lord Vader* (2015) by Debris depicts the iconic Darth Vader from the *Star Wars* movie series (see also Figure 2). Some works are game-like (e.g. *BOSSE* [2016] by Redcrab), while others directly refer to specific games, such as in the case of *100 points* (2015) by ilesj, portraying a character from *Wizard of Wor* (1983). Other popular-cultural references that we came across in our material were Internet memes, TV series, musicians and graffiti.

Conclusion

In this article, we have looked at PETSCII as both as a retro phenomenon and as an ongoing artistic practice. PETSCII can be easier to create than pixel art, chip music or demos, so these mosaic artworks give a low-threshold entry into the C-64 scene, especially after the emergence of new, accessible tools. The relative ease has created, among more seasoned works, many naïve pictures by beginners who want to give PETSCII a go. The recent renaissance can be attributed to high visibility competitions, lowered barriers to entry, social media and picture sharing sites, and also the general popularity of retrocomputing and related audiovisual styles.

The role of actual Commodore computers, the original hardware, has seen various changes throughout the decades. In addition to the journey from a popular product to a relic and finally the re-emergence as a cool retro platform, the machines' role in the creation of PETSCII graphics has, likewise, gone through notable transitions. In its heyday, the 8-bit computer itself was used in the creation, distribution and viewing of works. Nowadays, as most pictures are created with cross-platform tools and spread quickly on the Internet, original hardware mostly serves as a baseline that dictates what can be done. Even if the web browser is, in practice, the most common environment for viewing PETSCII art, competitions held at parties still aim for certain authenticity by showing the entries on real hardware.

We have shown works that clearly cater for nostalgia, like the line drawings on the iconic blue background of the C-64. On the other hand, photorealistic works do not play with the same kind of nostalgia. Quite the contrary: on a visual level the photorealistic PETSCII works seem to move away from the traditional look of PETSCII, and therefore some of its retrospective potentials. In extreme cases, the viewer cannot

even tell that the image is made in PETSCII. As such, the modes of production have been hidden by careful craftmanship typical of the demoscene. In addition to mere laid-back nostalgia, there is constant competition and push to "do the impossible" – PETSCII provides a tightly defined platform that sparks creativity and facilitates comparisons, as everybody operates within the same technical limitations.

Making our own editor was not only a means of artistic self-reflection through tool creation. As other people started creating works with it, the values and practices of the demoscene started manifesting themselves, as the tool had to facilitate easy participation in competitions. For instance, as it is still customary to deliver the works as executable (.prg) files, an export function soon became a necessity. The bitmap output format was also made to adhere to the preview guidelines of a major website. This way, the single artifact became valuable as a guiding and constraining device for our research, both as a way to anchor the research to a more bounded domain, and direct it towards the values within a community of creative artists.

When creating tools for the demoscene and hobbyists, conventions of technical and industrial development rarely fully apply. Although there are multiple new tools for easy PETSCII production, some leading artists feel it is significant to continue working with older tools or the original hardware itself. Just as appreciation for text art varies depending on the context, community and viewpoint, hobbyists and demosceners have their own cultural benchmarks regarding reputable tools, whether they are optimal or not. For the author of software and text art researcher, such observations foster a critical attitude towards technological platforms as highly developed entities, and rather reinforces the role of tools as cultural products embedded in a larger frame of reference.

As the interest towards the historicity of home computer media increases in (media) arts studies, we think it is worthwhile to explore and examine a phenomenon such as text art in all its richness. This includes tool creation and adoption by communities with their competitions, art exhibitions and other publication channels. Combining the analysis of existing works with creative activities produces experiential and analytical knowledge, which can be utilized in both further artistic endeavors and academic research.

Acknowledgements

We thank the Academy of Finland for funding the *Centre of Excellence in Game Culture Studies* research program.

References

Bagnall, B. (2005). *On the edge: The spectacular rise and fall of Commodore*. Winnipeg: Variant Press.

Carlsson, A. (2017). Beyond encoding: A critical look at the terminology of text graphics. *WiderScreen*, *20*(1–2). Retrieved from http://widerscreen.fi/numerot/2017–1-2/beyond-encoding-a-critical-look-at-the-terminology-of-text-graphics/

Carlsson, A., & Miller, A.B. (2012). Future potentials for ASCII art. In: K. Zreik, & R. Gareus (Eds.), *Postdigital art – Proceedings of the 3rd computer art congress* (pp. 13–24). Paris: Europia.

Danet, B. (2003). Pixel patchwork: 'Quilting in time' online. *Textile*, *1*(2): 118–143.

Franke, H. W. (1971). *Computer graphics – computer art*. London: Phaidon Press.

Giraldo Arteaga, M. (2014, September 3). Generative ebook covers [Blog post]. Retrieved from https://www.nypl.org/blog/2014/09/03/generative-ebook-covers

Hardagon, M. (2011). *Like city lights, receding: ANSi artwork and the digital underground 1985–2000*. (Unpublished master's thesis.) Concordia University.

Heikkinen, T., & Reunanen. M. (2017). *Rock, joka tiesi liikaa* [Rock who knew too much]. *WiderScreen*, *20*(1–2). Retrieved from http://widerscreen.fi/numerot/2017–1-2/rock-tiesi-liikaa/

Higgins, H. (2009). *The grid book*. Cambridge, MA: MIT Press.

Lindsay, C. (2003). From the shadows: Users as designers, producers, marketers, distributors and technical support. In: N. Oudshoorn & T. Pinch (Eds.), *How users matter: The co-construction of users and technologies* (pp. 29–50). Cambridge, MA: MIT Press.

Maher, J. (2012). *The future was here: The Commodore Amiga*. Cambridge, MA: MIT Press.

Manovich, L. (2001). *The language of new media*. Cambridge, MA: MIT Press.

Meyers, R. (2017). Keys of fury – Type in beyond the scrolling horizon. *WiderScreen*, *20*(1–2). Retrieved from http://widerscreen.fi/numerot/2017–1-2/keys-of-fury-type-in-beyond-the-scrolling-horizon/

Moe, H., & Van den Bulck, H. (Eds.). (2016). *Teletext in Europe: From the analog to the digital era*. Gothenburg: Nordicom.

Montfort, N., Baudoin, P., Bell, J., Bogost, I., Douglass, J., Marino, M.C., Mateas, M., Reas, C., Sample, M., & Vawter, N. (2012). *10 PRINT CHR$(205.5+RND(1)); : GOTO 10*. Cambridge, MA: MIT Press.

Musturi, T. (2017). Unplanned blocky puzzles – Creating PETSCII for the Commodore 64. *WiderScreen*, *20*(1–2). Retrieved from http://widerscreen.fi/numerot/2017–1-2/unplanned-blocky-puzzles-creating-petscii-for-the-commodore-64/

Paul, L.J. (2013). Text-mode and the live PETSCII animations of Raquel Meyers: Finding new meaning through live interaction. *Leonardo Electronic Almanac*, *19*(3). Retrieved from http://www.leoalmanac.org/wp-content/uploads/2013/11/LEAVol19No3-Paul.pdf

Polanyi, M. (1966). *The tacit dimension*. Chicago, IL: University of Chicago Press.

Reas, C., & Fry, B. (2014). *Processing: A programming handbook for visual designers and artists* (2nd ed.). Cambridge, MA: MIT Press.

Saarikoski, P., Suominen, J., & Reunanen, M. (2017). Pac-man for the VIC-20: Game clones and program listings in the emerging Finnish home computer market. *Well Played*, *6*(2): 7–31.

Schön, D. (1991). *The reflective practitioner: How professionals think in action*. Aldershot: Ashgate Publishing.

Silvast, A., & Reunanen, M. (2014). Multiple users, diverse users: Appropriation of the personal computer by demoscene hackers. In: G. Alberts, & R. Oldenziel (Eds.), *Hacking Europe: From computer cultures to demoscenes* (151–163). Berlin: Springer.

Xu, X., Zhang, L., & Wong, T. (2010). Structure-based ASCII art. *ACM Transactions on Graphics, 29*(4): 52:1–52:10.

Abstract

PETSCII is the built-in character set used on 8-bit Commodore computers, such as the PET, C-64 and Plus/4. The character set and the BASIC environment provided an entry point to rudimentary graphic editing for a generation of hobbyists. Over the years PETSCII has been used for a variety of creative purposes: for example games, demos, telecommunications, videos, books and fine art have been created using the graphical symbols. Through the analysis of existing art works and our own hands-on project – a graphics editor – we dig deeper into the specific properties of the character set. In this article we show how the fixed symbols set the frame for what is possible and how, on the other hand, they provide grounds for creative experimentation, display of skill and recognizable styles.

Keywords: text art, digital culture, retrocomputing

Authors

Markku Reunanen (markku.reunanen@iki.fi), PhD, LicSc, is a senior lecturer at Aalto ARTS and a postdoctoral researcher at the University of Turku. His research interests include digital culture, retrocomputing and digital cultural heritage.

Tero Heikkinen (tero.heikkinen@uniarts.fi), DA, is a postdoctoral researcher at the Academy of Fine Arts in the University of the Arts Helsinki. He studies the uses of media platforms and tools in art research domains.

Anders Carlsson (info@goto80.com), MSc, is an artist and independent researcher interested in the materials, artistic practices and subcultures of old new media.

Appendix: List of Analyzed Works

100 points by ilesj (2015)	*Good Luck* by The USER (2016)	*Old Geek* by Salvo Bee (2017)
Asterix bei den Pikten by Domspitze (2013)	*Han som reiste* by 8R0TK4$T3N (2013)	*Pikachu* by Mary B (2013)
Black Dynamite by Cipi (2013)	*Hunter in the Shallows* by Hammerfist (2013)	*Pipe of Picard* by Electric (2016)
BOSSE by Redcrab (2016)	*Keys of Fury* by AcidT* (2016)	*Rebus 100%* by q0w (2013)
C64 Scene History by G-Fellow (2013)	*Kuha ois entry* by Totuus (2015)	*Setae Neuvoo* by Cock Norris (2015)
Coltrane by Electric (2015)	*Lord Vader* by Debris (2015)	*Space Diva* by Wile Coyote (2017)
Daft Peek by Man (2013)	*Mantis Compo Version* by Slayer Graphics (2013)	*Submerged* by Terwiz (2015)
Directory Karate + by Scott (2017)	*Midnight Hour* by Hammerfist (2013)	*Triceratopetscii* by ptoing (2016)
Displosion by Logiker (2017)	*MUN EKA PETSCII* by Hi-stack (2013)	*Variable Medusa* by Rexbeng (2013)
Full Horizontal Directory Logo by Goat (2017)	*New York* by daimansion (2013)	*Vladimirovich* by Otium (2015)
Gary by Mermaid (2013)	*Ninja Enters the Battle* by ilesj (2013)	*We Are All Ejected* by Archmage (2016)
Gnome - Sharpened by Shine (2017)	*Nothing but PETSCII* by Genesis Project (2016)	