ChatGPT

ASCII/ANSI Art Editors and Winamp-Style Visualizer Preset Tools

Master Catalog of Tools

Below is a compiled inventory of notable tools in two domains: **(A) ASCII/ANSI/Text Art Editors & Toolchains** (including character art formats like PETSCII, teletext, etc.), and **(B) Winamp-Style Audio Visualization Preset Ecosystems**. Each tool entry is categorized and annotated with key attributes (platforms, license, status, etc.). This catalog is delivered in both CSV and JSON formats for further analysis.

CSV Schema (columns):

`name, category, primary_formats, license, open_source, repo_url, homepage_url, platforms, status, first_release, last_release, price, core_features, notable_limitations, export_options, import_options, web_rendering_support, community_metrics, docs_quality, examples_or_screenshots, notes`

```
name, category, primary_formats, license, open_source, repo_url, homepage_url,
platforms, status, first_release, last_release, price, core_features,
notable_limitations, export_options, import_options, web_rendering_support,
community_metrics, docs_quality, examples_or_screenshots, notes
TheDraw, ansi-editor, [ANS, ASC, BIN], Proprietary (Shareware), false, (n/a),
(n/a), DOS, historical (discontinued), 1986, 1993, free (abandonware), "ANSI/
ASCII editing; animations; built-in font library (TDF); 25-100 line canvas
limit", "DOS only; 100-line file limit [1]; no native Unicode", "ANS, ASC, BIN
(TheDraw fonts), BSAVE", "n/a (draw from scratch only)", "n/a (text-mode only)",
"Widely used in BBS era; Basis for many later tools [2]", "okay (v4.63 manual
available)", (see 16colo.rs archive), "Pioneering ANSI editor; defined many
standard features [2]"
ACiDDraw, ansi-editor, [ANS, ASC, BIN], Freeware, false, (n/a), [3], DOS (Win9x
via DOS), historical (abandoned), 1994, ~1998, free, "Extended canvas up to 1000
lines [4] [5]; iCE color support; ANSI/ASCII modes", "DOS only; requires DOSBox on
modern OS; UI dated", "ANS, ASC, BIN (fonts)", "n/a", "n/a", "Popular in
artscene 90s; included in ACiD art packs", "poor (help files in distribution)",
(screenshots at roysac.com), "Successor to TheDraw, removed 100-line limit [4];
used by ACiD Art group"
DarkDraw, ansi-editor, [ANS, ASC], Freeware, false, (n/a), [6] [7], DOS,
historical (abandoned), 1994, 1994, free, "Extended memory for 3000+ lines [8];
ASM optimized; ANSI viewer included (HavokView)", "DOS only; keyboard issues on
non-IBM compatibles [9]", "ANS (up to 3000 lines), BIN", "ANS import for
editing", "n/a", "Niche usage (HaVoK group) [10]; now a curiosity via DOSBox",
"okay (included .DOC file)", (artifact downloads available), "Notable for
```

pushing text art length limits (3k lines) [11] [8] "
PabloDraw, ansi-editor, [ANS, PCB, BIN, RIP], MIT, true, [12], [13], Windows,macOS,Linux, active, ~2000, 2017-06-30, free, "GUI ANSI/ASCII editor & viewer; multi-user collaborative drawing; supports Amiga ANSI (Topaz font) [14]; RIP script vector art", "No Unicode (CP437 only); development has slowed after open-sourcing", "ANS, PCB, BIN, RIP, SAUCE metadata", "ANS/PCB import (merge or edit), RIP import", "Partial (exports HTML via ansilove library)", "Over 370k downloads [15]; used on 16colo.rs BBS art community", "good (online manual, active forums)", (homepage has screenshots), "De-facto modern ANSI editor; cross-platform .NET rewrite with networking [16]"
Moebius, ansi-editor, [ANS, ASC, XB/XBIN], Apache-2.0, true, [17], [18], Windows,macOS,Linux, maintained (WIP), 2019, 2023, free, "Modern ANSI/ASCII art GUI editor; unique half-block brush tool [19]; layers, block/fluid drawing modes", "No GUI animation support; Unicode limited to CP437 range (for now)", "ANS, ASCII (.TXT), XBIN (view/save) [20]", "ANS, XBIN import; custom fonts import (XBIN) [21] [22]", "No built-in web export (use Ansilove or viewer)", "840★ on GitHub [23]; community: Blocktronics collective", "okay (README, GitHub issues for help)", (GitHub page shows UI), "Open-source editor by blocktronics; aims to supersede PabloDraw with Photoshop-like workflow [18]"
MoebiusXBIN, unicode-art-editor, [XBIN, ANS, TXT], GPL-3.0, true, [24], [24], Windows,macOS,Linux, active, 2020, 2025-07-30, free, "Fork of Moebius adding *custom fonts & palettes* in XBIN [20] [22]; 2400+ font library, palette browser, right-to-left text support [22]", "Complex UI for newcomers; tied to XBIN format for full feature use", "XBIN (with embedded fonts/colors), ANS, TXT, PNG fonts", "ANS import, font PNG import/export [25]", "n/a (desktop only, but XBIN viewer needed)", "Used by experimental ASCII font artists [26]; niche but growing", "good (extensive tutorial included) [27]", (tutorial images on blog), "Extends text-mode art beyond legacy limits with new fonts/colors [20] [26]"
Icy Draw, ansi-editor, [ANS, ASC, BIN, XBIN], GPL-3.0, true, [28] [29], [30], Windows,macOS,Linux (OpenGL), active, 2023, 2023, free, "Cross-platform ANSI/ASCII editor (successor to Mystic Draw) with **layers** and multi-canvas tabs [31]; OpenGL-accelerated; supports XBIN, Artworx, iCE fonts", "Very new (some bugs); no native terminal-mode usage", "ANS, XBIN, BIN (multiple formats: PCB, TND, ADF) [32]", "Imports legacy formats (ANSI, Avatar, PCB) [32]; TheDraw .TDF fonts support (planned) [33]", "Planned: HTML/PNG via ansilove (not yet integrated)", "New release announced on r/bbs with positive feedback [34]; small but enthusiastic user base", "okay (basic docs and Reddit Q&A)", (YouTube “Let’s ANSI - Icy Draw” demo), "Modern take on ANSI editing aiming for heavy-duty use (OpenGL, fast rendering) [35]"
Durdraw, ansi-editor, [ANS, TXT, DFD (Durdraw)], BSD-3-Clause, true, [36] [37], [38], Linux,macOS (Terminal), active, 2017, 2023, free, "Terminal-based text art studio; supports **frame-based animations** [39]; 16-color and 256-color (xterm) modes; Unicode (UTF-8) and CP437 mixing [39]; mouse support in terminal", "CLI UI (learning curve); no GUI; PNG/GIF export only in 16-color mode [40] [41]", "ANS (SAUCE), JSON, DFD (internal), HTML, PNG/GIF [40]", "ANS import (CP437 Unicode conversion) [39]; image→ANSI conversion (external scripts)", "Yes – HTML output, plus requires Ansilove for PNG/GIF [40]", "HN discussion praised its modern

terminal approach [42]; niche Unix enthusiasts [43]", "good (README with usage, man pages)", (GitHub gallery, demos available), "Blends retro ANSI editing with modern terminal capabilities (truecolor, theming) [37] [41]"
TetraDraw, ansi-editor, [ANS, TND], GPL-2.0, true, [44], (SourceForge), Linux (ncurses), archived, 2003, 2004, free, "First full-featured ANSI editor on Unix [44]; **real-time multi-user drawing over TCP/IP** [45]; curses interface", "Unix text-mode only; no updates since early 2000s", "ANS, TND (TundraDraw) [46]", "n/a (manual editing only)", "n/a (text terminal output)", "Important historically for Linux BBS scene [44]; superseded by newer tools", "okay (man page, docs included)", "(n/a)", "Enabled Linux users to create ANSI without DOS emulation [47]"
TundraDraw, ansi-editor, [ANS, TND], GPL-2.0, true, [48], [48], Windows, Linux (Qt), dormant, 2002, 2009, free, "Qt-based GUI ANSI editor; cross-platform; introduced its own .TND format (sauce-like)", "Beta quality (0.5) [49]; development halted ~2010", "ANS, TND, BIN, ADF", "Imports ANSI, XBin, Artworx formats", "No (desktop GUI only)", "Allowed Win/Linux art creation in 2000s [50]; kept scene alive post-DOS [50]", "poor (sparse readme)", (RoySAC hosts download [49]), "Referenced as tool that sustained late-2000s ANSI art [50]"
JavE (Java ASCII Versatile Editor), ascii-editor, [TXT, ASC, XML], Freeware, false, [51], [51], Cross-platform (Java), historical, 2001, 2013, free, "General ASCII art editor (focus on line-drawing & diagrams) [52]; GUI with drawing tools, text effects, image-to-ASCII converter", "No color/ANSI support (monochrome only); Java app UX dated", "TXT (ASCII), proprietary .jave (XML)", "Image import to ASCII (bw)", "Exports HTML with ASCII art", "Once considered *“the BEST AA editor”* for Shift-JIS/ASCII art [53]", "okay (bundled help)", "(screenshots on jave.de)", "Popular in early 2000s especially for Shift-JIS art; now legacy"
Monodraw, ascii-editor, [TXT, PNG, SVG], Proprietary, false, (n/a), [54], macOS, maintained (commercial), 2015, 2023, paid ($49.99), "**Polished Mac-native ASCII art designer** (diagrams, flowcharts, layouts) [54]; shape tools, connectors, real-time table drawing; exports to text, PNG, SVG", "Mac only; closed source; relatively expensive", "TXT, PNG, SVG, PDF, HTML", "Import ASCII text files", "N/A (exports as image or HTML only)", "Niche professional user base (developers, designers) [55]; positive reviews on HN [56]", "good (extensive manual)", (official site showcases), "Sets high bar for UX in ASCII editing; unique diagramming focus beyond BBS art"
REXPaint, ascii-editor, [XP, TXT, ANS, PNG], Freeware, false, (n/a), [57], Windows (runs via Wine on *nix) [58], maintained, 2013, 2023, free, "Roguelike-centric ASCII/ANSI editor [59] with **multi-layer support** [60], 24-bit color, shape drawing tools; widely used for game art/mockups", "No native Linux/macOS build (Windows only, but Wine OK) [61]; closed source", "XP native format, ANS, TXT, CSV, PNG (image export)", "Import XP files, plain text", "Exports PNG (with transparency), ANS (16-color), TXT", "Popular in roguelike dev community [62] (used in Cogmind dev); ~10k downloads", "good (PDF manual [63])", (gallery on website), "Combines retro text art and game design needs (e.g. JSON export to game engines) [63]"
ASCII Art Paint, unicode-art-editor, [TXT, PNG], GPL-3.0, true, [64], [65], Web (browser), active, 2018, 2023, free, "Web-based ASCII graphics editor [66]; draws

with customizable symbol palettes; supports converting images to ASCII; live preview", "Browser-based only; limited to monochrome or simple color (no ANSI escapes)", "TXT, HTML, PNG (image of ascii)", "Import images (for conversion)", "Canvas element for live HTML preview", "Open source on GitHub (release 11) [67]; modest adoption via itch.io", "okay (some tutorials via itch.io)", (online demo available), "Modern browser tool bridging pixel-art and text art creation [66]"
ASCIIFlow, ascii-editor, [TXT, HTML], Proprietary (free), false, (n/a), [68], Web, active, 2011, 2021, free, "Simple online ASCII diagramming tool; grid canvas with lines/boxes drawing; Google Drive save integration [69]", "No color support; limited character set (for diagrams only)", "TXT, HTML (pre-formatted)", "Import ASCII via copy-paste", "Embeddable HTML (monospace text)", "Popular for quick diagramming in code comments [70]", "okay (minimal help on site)", "(live site)", "Lightweight, focuses on schematics/box art rather than full ANSI art"
Figlet, converter, [TXT], BSD-3-Clause, true, (n/a), [71], Cross-platform (CLI), active, 1991, 2018, free, "Classic text-to-ASCII-art font generator [71]; dozens of font styles (block letters, scripts, etc.)", "Not an interactive editor (CLI only); output is plain ASCII (no color)", "TXT (ASCII art text output)", "Takes input text strings", "n/a (text output only)", "Commonly used to generate title text for ASCII art pieces", "good (man page)", "(n/a)", "Useful as plug-in for editors (e.g. generate ASCII logo text) [71]"
TOIlet, converter, [TXT, ANSI], GPL-3.0, true, (n/a), (GNU Savannah), Cross-platform (CLI), maintained, 2007, 2014, free, "Enhanced Figlet-like tool that supports ANSI color output and Unicode fonts (e.g. emoji)", "Same limitations as Figlet: not WYSIWYG editor", "TXT, ANS (colored text output)", "Takes text input", "n/a (console colored text)", "Fills gap by adding color and Unicode to figlet banners", "okay (info pages)", "(n/a)", "Can be integrated into modern text art workflows for fancy colorful text banners"
boxes, converter, [TXT], GPL-2.0, true, (n/a), (packages.debian.org/boxes), Cross-platform (CLI), maintained, 1999, 2012, free, "CLI tool to draw ASCII art boxes/frames around text; many predefined styles (balloons, designs)", "Not an editor; limited to framing existing text", "TXT (text with box outline)", "Takes input text (stdin or file)", "n/a", "Often used in combination with figlet output to create banners", "okay (man page)", "(n/a)", "Could be integrated into GUI editors for one-click borders"
Ansilove (& libansilove), renderer, [PNG, GIF, HTML], BSD-2-Clause, true, [40], [72], Cross-platform (C library & CLI), active, 2011, 2023, free, "**ANSI/ASCII art to image converter** [40] supporting ANSI, PCB, BIN, GIF animations; retains SAUCE metadata; used by 16colo.rs for gallery images", "Not an editor; conversion only", "PNG, GIF, HTML (from text input)", "Input: .ANS, .PCB, .BIN, .ADF, .IDF etc.", "Embeddable Web (Ansilove.js exists for browser)", "Standard tool in scene for high-quality ANSI to PNG rendering [40]", "good (docs on ansilove.org)", "(n/a)", "Ensures text art is viewable on web/forums without special fonts (pixel-perfect conversion) [40]"
iNFekt NFO Viewer, renderer, [ANS, NFO, DIZ], BSD-2-Clause, true, (GitHub), Windows, Linux, active, 2010, 2021, free, "Dedicated viewer for ANSI/NFO text art with accurate colors, triple column view, zoom, and export to image", "View-only (no editing); Windows GUI, Linux (Qt) version available", "n/a (viewer

only)", "n/a", "Export to PNG image", "Popular for viewing scene .NFO files (releases, NFOs) with proper oldskool fonts", "good (help file)", "(screenshot on GitHub)", "Fulfills need to read text art easily on modern OS without corrupt rendering"
Teletext Editor (Edit.TF), teletext-editor, [TTI, TTX, PNG], MIT, true, [73], [73], Web, active, 2016, 2023, free, "Browser-based **Teletext art** editor; authentic Mode 7 editing grid; supports level 1 teletext (24x40, 8-color) [74]", "Narrow use (teletext only); limited resolution and color palette", "EP1, TTI, TTX (teletext page formats), PNG", "Import teletext format or PNG", "Canvas in browser (WebGL for preview)", "Revival of teletext art community, used in teletext art comps", "okay (tutorial available) [75]", "(online tool)", "Allows creation of art for teletext services and emulators; niche but active scene"
PETSCII Editor (Petmate), petscii-editor, [PETSCII, PRG, PNG], MIT, true, [76] [77], [78], Windows,macOS,Linux, active, 2017, 2019, free, "Cross-platform **Commodore 64 PETSCII** art editor; multiple screens, undo/redo [79], custom charsets, exports to C64 formats (.prg, .asm) [77], JSON, PNG", "PETSCII-specific (commodore glyphs only); last update 2019", "PETSCII (.prg, .asm, .64c), PNG, Petmate JSON", "Import .d64 disk art, .png to PETSCII (pixel-match) [80]", "No (desktop only)", "Widely used in C64 demoscene for text graphics [81]; recommended on r/c64 [81]", "good (README, shortcuts) [82]", "(screenshot in docs)", "Modern tool keeping PETSCII art alive, with conveniences like multicolor mode and JSON export for integration [83]"
Marq's PETSCII Editor, petscii-editor, [PRG, PNG, ANS], GPL-2.0, true, [84], [85], Windows,macOS,Linux, archived, 2011, 2015, free, "Original multi-platform PETSCII editor by Marq; supports C64 and VIC-20 fonts, animations [86]", "Superseded by Petmate (no longer updated)", "PRG, standalone .exe, PNG, anim GIF", "Imports PETSCII prg, images", "PNG export", "Contained a PETSCII art gallery online [87]", "okay", "(gallery online)", "Pioneered modern PETSCII editing; provided inspiration for newer tools"
VSXu (VSX Ultra), visualizer-editor, [VSXu preset (.vsxu)], GPL-3.0, true, (GitHub), Windows,Linux, semi-maintained, 2009, 2017, free, "**Standalone music visualization system** with node-based preset editor; not Milkdrop-compatible (parallel ecosystem)", "Steeper learning curve (modular synth-like); smaller preset library than Milkdrop", "VSXu preset files (graph definitions)", "Import images/shaders as nodes", "Exports visuals via SDK (for integration)", "Used by some VJs and demosceners as open alternative to Winamp AVS/Milkdrop", "okay (docs & tutorials exist)", (editor UI screenshot), "Shows what a modern visual preset IDE can be (graphical patching of effects), but not widely adopted outside niche"
Winamp AVS (Advanced Visualization Studio), visualizer-preset-editor, [AVS preset (.avs)], Proprietary (formerly BSD-like for SDK), *partial*, [88] [89], Win32, historical (open-source community fork), 1998, 2005 (2.82), free, "**Classic Winamp visualization plugin** featuring a built-in graphical preset editor (mix of effects, DM codes) [90]; huge library of community presets (1000s on Winamp forums, DeviantArt [91])", "No official updates since 2005; performance issues on modern OS; presets require old Winamp or ports", ".AVS (binary preset), .APE (plugin effects)", "Import user presets packages", "No (real-time only; screenshots possible)", "Thriving 2000s community (Visbot archived 15k

posts [92]; AVSociety on DeviantArt) but now niche", "okay (wiki and old forum guides) [93]", "(AVS preset packs gallery) [94]", "Notable for visual programming approach to music visuals; groundwork for later shader-based systems"
MilkDrop (Winamp MilkDrop 2.x), visualizer-preset-editor, [Milk (.milk) preset], BSD-2-Clause (Milkdrop 2 OSS release) [95], *partial* (projectM), [96], Win32 (Winamp plugin), legacy (OSS code via projectM), 2001, 2007 (v2.25), free, "Hardware-accelerated music visualization plugin for Winamp [97]; **preset-driven shader effects** with audio-responsive equations; ~50k community presets created [98]", "Official dev halted in 2007; editing presets requires manual code tweaking or in-plugin menu [99]", ".milk (preset files), Milkdrop preset pack (.zip)", "Presets can be edited live in Winamp (text UI) [99]; import old MD1 presets", "Screenshots, AVI via Winamp video out", "Iconic visualizer – still beloved; revived via projectM & Butterchurn on modern platforms [100]", "good (extensive authoring guides online) [101]", "(Geisswerks tutorial, Pawel Porwisz guide) [101]", "Benchmarked for all modern visualizers; vibrant preset scene (73k pack available [98])"
projectM, visualizer-player, [Milk preset, PRJM], LGPL-2.1 [102], true, [103], Cross-platform (C++ library + apps), active, 2003, 2024, free (GPL app/paid mobile), "Open-source reimplementation of MilkDrop** [103] as a cross-platform library and apps; runs MilkDrop presets unmodified; integrations for VLC, Kodi, foobar2000, etc.", "Historically lagged behind MilkDrop on newer preset features (now catching up); preset authoring still code-oriented", ".milk, .prjm (projectM new format)", "Import MilkDrop presets and playlists", "Export still images, video capture in some frontends", "Most advanced FOSS visualizer [103]; large user base via app stores and media players [104]", "good (active wiki, Discord support) [105]", "(video demos available)", "Ensures MilkDrop’s legacy on modern systems (Win/Mac/Linux, even Android TV) [106] [107]; constant improvements (v4+ supports modern OpenGL, JSON preset editing, etc.)"
Butterchurn, visualizer-player, [Milk preset, JSON], MIT, true, [108], [109], Web (JavaScript/WebGL2), active, 2018, 2023, free, "WebGL **MilkDrop 2 implementation** [109] – brings Winamp visuals to browsers; high-performance (WASM-compiled preset shaders) [110]; used in Webamp and others", "Limited to MilkDrop preset compatibility; no GUI editor (presets edited as text JSON)", ".milk (loaded), JSON (internal compiled form)", "Milk presets (v1 & v2) load via JS, optional conversion to JSON for speed [108]", "Canvas/WebGL output embeddable in any webpage", "Widely adopted for web music players and streams (audio-reactive web art) [100]; active dev community on GitHub", "good (API documentation)", "(butterchurnviz.com demo) [109]", "Extends MilkDrop’s reach to modern web and cross-device usage [100]; even used in Steam, Cider apps [111]"
WebVS (Web AVS), visualizer-player, [AVS preset], MIT, true, [112], [112], Web (JS/WebGL), inactive, 2014, 2015, free, "Experimental JavaScript/WebGL reimplementation of Winamp AVS [112]; could run many classic AVS presets in browser", "Project incomplete – not all AVS effects supported; last updated 2015", ".avs presets (limited support)", "Import AVS presets (some may not work fully)", "Canvas/WebGL output for web embed", "Proof-of-concept level; demonstrated AVS on the web for nostalgia", "poor (sparse README)", "(GitHub repo)", "Showed that AVS’s visual logic could be ported to JS, but never fully realized; superseded by modern shader viz"

```
Winamp Advanced Visualization Studio DX (vis_avs_dx), visualizer-player, [AVS
preset], Apache-2.0, true, [113], [113], Windows, active (community fork), 2020,
2021, free, "Updated build of classic AVS with **DirectX rendering** (hardware
accel)[113] and bug fixes; allows AVS to run on modern Windows and at higher
framerates", "No new features beyond rendering pipeline; still Windows-only and
requires Winamp/WACUP", ".avs", "AVS presets only", "n/a (real-time only)",
"Keeps vast archive of AVS presets usable today (works with WACUP, etc.)", "okay
(community forums)", "(GitHub vis_avs_dx)", "Important maintenance update
extending life of AVS visual content"
AVS Plugin (MusicBee etc.), visualizer-player, [AVS preset], Proprietary,
false, [114], [114], Windows, maintained, 2018, 2022, free, "Wrapper allowing
**Winamp AVS presets in MusicBee player**[114] (and some other players); leverages
community AVS builds", "Specific to MusicBee (closed-source plugin); relies on
legacy code", ".avs", "AVS presets only", "n/a", "Helps preserve old presets
outside Winamp ecosystem", "okay (MusicBee forum)", "(MusicBee addon page)",
"Indicative of continued interest in classic visualizations on new platforms"
```

(Note: In CSV above, some fields containing commas are quoted for CSV correctness. `open_source` *marked "*partial*" where source code was later opened or reimplemented via another project.)*

The same data is provided in JSON format for structured consumption:


```
[
  {
    "name": "TheDraw",
    "category": "ansi-editor",
    "primary_formats": ["ANS", "ASC", "BIN"],
    "license": "Proprietary (Shareware)",
    "open_source": false,
    "repo_url": null,
    "homepage_url": null,
    "platforms": "DOS",
    "status": "historical (discontinued)",
    "first_release": 1986,
    "last_release": 1993,
    "price": "free (abandonware)",
    "core_features": "ANSI/ASCII editing; animations; built-in font library
(TDF); 25-100 line canvas limit",
    "notable_limitations": "DOS only; 100-line file limit[1]; no native
Unicode",
    "export_options": "ANS, ASC, BIN (TheDraw fonts), BSAVE",
    "import_options": "n/a (draw from scratch only)",
    "web_rendering_support": "n/a (text-mode only)",
    "community_metrics": "Widely used in BBS era; Basis for many later tools
[2]",
    "docs_quality": "okay (v4.63 manual available)",
```

```
    "examples_or_screenshots": "(see 16colo.rs archive)",
    "notes": "Pioneering ANSI editor; defined many standard features [2]"
  },
  {
    "name": "ACiDDraw",
    "category": "ansi-editor",
    "primary_formats": ["ANS", "ASC", "BIN"],
    "license": "Freeware",
    "open_source": false,
    "repo_url": null,
    "homepage_url": " [3] ",
    "platforms": "DOS (Win9x via DOS)",
    "status": "historical (abandoned)",
    "first_release": 1994,
    "last_release": "~1998",
    "price": "free",
    "core_features":
"Extended canvas up to 1000 lines [4] [5]; iCE color support; ANSI/ASCII modes",
    "notable_limitations": "DOS only; requires DOSBox on modern OS; UI dated",
    "export_options": "ANS, ASC, BIN (fonts)",
    "import_options": "n/a",
    "web_rendering_support": "n/a",
    "community_metrics": "Popular in artscene 90s; included in ACiD art packs",
    "docs_quality": "poor (help files in distribution)",
    "examples_or_screenshots": "(screenshots at roysac.com)",
    "notes": "Successor to TheDraw, removed 100-line limit [4]; used by ACiD Art
group"
  },
  {
    "name": "DarkDraw",
    "category": "ansi-editor",
    "primary_formats": ["ANS", "ASC"],
    "license": "Freeware",
    "open_source": false,
    "repo_url": null,
    "homepage_url": " [6] ",
    "platforms": "DOS",
    "status": "historical (abandoned)",
    "first_release": 1994,
    "last_release": 1994,
    "price": "free",
    "core_features": "Extended memory for 3000+ lines [8]; ASM optimized; ANSI
viewer included (HavokView)",
    "notable_limitations": "DOS only; keyboard issues on non-IBM compatibles
[9]",
    "export_options": "ANS (up to 3000 lines), BIN",
    "import_options": "ANS import for editing",
    "web_rendering_support": "n/a",
```

```
    "community_metrics": "Niche usage (HaVoK group) [10]; now a curiosity via DOSBox",
    "docs_quality": "okay (included .DOC file)",
    "examples_or_screenshots": "(artifact downloads available)",
    "notes": "Notable for pushing text art length limits (3k lines) [11] [8]"
  },
  {
    "name": "PabloDraw",
    "category": "ansi-editor",
    "primary_formats": ["ANS", "PCB", "BIN", "RIP"],
    "license": "MIT",
    "open_source": true,
    "repo_url": " [12] ",
    "homepage_url": " [13] ",
    "platforms": "Windows, macOS, Linux",
    "status": "active",
    "first_release": 2000,
    "last_release": "2017-06-30",
    "price": "free",
    "core_features": "GUI ANSI/ASCII editor & viewer; multi-user collaborative drawing; supports Amiga ANSI (Topaz font) [14]; RIP script vector art",
    "notable_limitations": "No Unicode (CP437 only); development has slowed after open-sourcing",
    "export_options": "ANS, PCB, BIN, RIP, SAUCE metadata",
    "import_options": "ANS/PCB import (merge or edit), RIP import",
    "web_rendering_support": "Partial (exports HTML via ansilove library)",
    "community_metrics": "Over 370k downloads [15]; used on 16colo.rs BBS art community",
    "docs_quality": "good (online manual, active forums)",
    "examples_or_screenshots": "(homepage has screenshots)",
    "notes": "De-facto modern ANSI editor; cross-platform .NET rewrite with networking [16]"
  },
  {
    "name": "Moebius",
    "category": "ansi-editor",
    "primary_formats": ["ANS", "ASC", "XBIN"],
    "license": "Apache-2.0",
    "open_source": true,
    "repo_url": " [17] ",
    "homepage_url": " [18] ",
    "platforms": "Windows, macOS, Linux",
    "status": "maintained (WIP)",
    "first_release": 2019,
    "last_release": 2023,
    "price": "free",
    "core_features": "Modern ANSI/ASCII art GUI editor; unique half-block brush tool [19]; layers, block/fluid drawing modes",
```

```
    "notable_limitations": "No GUI animation support; Unicode limited to CP437 range (for now)",
    "export_options": "ANS, ASCII (.TXT), XBIN (view/save) [20]",
    "import_options": "ANS, XBIN import; custom fonts import (XBIN) [21] [22]",
    "web_rendering_support": "No built-in web export (use Ansilove or viewer)",
    "community_metrics": "840★ on GitHub [23]; community: Blocktronics collective",
    "docs_quality": "okay (README, GitHub issues for help)",
    "examples_or_screenshots": "(GitHub page shows UI)",
    "notes": "Open-source editor by blocktronics; aims to supersede PabloDraw with Photoshop-like workflow [18]"
  },
  {
    "name": "MoebiusXBIN",
    "category": "unicode-art-editor",
    "primary_formats": ["XBIN", "ANS", "TXT"],
    "license": "GPL-3.0",
    "open_source": true,
    "repo_url": " [24]",
    "homepage_url": " [24]",
    "platforms": "Windows, macOS, Linux",
    "status": "active",
    "first_release": 2020,
    "last_release": "2025-07-30",
    "price": "free",
    "core_features": "Fork of Moebius adding custom fonts & palettes in XBIN [20] [22]; 2400+ font library, palette browser, right-to-left text support [22]",
    "notable_limitations": "Complex UI for newcomers; tied to XBIN format for full feature use",
    "export_options": "XBIN (with embedded fonts/colors), ANS, TXT, PNG fonts",
    "import_options": "ANS import, font PNG import/export [25]",
    "web_rendering_support": "n/a (desktop only, but XBIN viewer needed)",
    "community_metrics": "Used by experimental ASCII font artists [26]; niche but growing",
    "docs_quality": "good (extensive tutorial included) [27]",
    "examples_or_screenshots": "(tutorial images on blog)",
    "notes": "Extends text-mode art beyond legacy limits with new fonts/colors [20] [26]"
  },
  {
    "name": "Icy Draw",
    "category": "ansi-editor",
    "primary_formats": ["ANS", "ASC", "BIN", "XBIN"],
    "license": "GPL-3.0",
    "open_source": true,
    "repo_url": " [28]",
    "homepage_url": " [30]",
```

```
    "platforms": "Windows, macOS, Linux (OpenGL)",
    "status": "active",
    "first_release": 2023,
    "last_release": 2023,
    "price": "free",
    "core_features": "Cross-platform ANSI/ASCII editor (successor to Mystic
Draw) with layers and multi-canvas tabs [31]; OpenGL-accelerated; supports XBIN,
Artworx, iCE fonts",
    "notable_limitations": "Very new (some bugs); no native terminal-mode
usage",
    "export_options": "ANS, XBIN, BIN (multiple formats: PCB, TND, ADF) [32]",
    "import_options": "Imports legacy formats (ANSI, Avatar, PCB) [32];
TheDraw .TDF fonts support (planned) [33]",
    "web_rendering_support": "Planned: HTML/PNG via ansilove (not yet
integrated)",
    "community_metrics": "New release announced on r/bbs with positive feedback
[34]; small but enthusiastic user base",
    "docs_quality": "okay (basic docs and Reddit Q&A)",
    "examples_or_screenshots": "(YouTube “Let’s ANSI - Icy Draw” demo)",
    "notes": "Modern take on ANSI editing aiming for heavy-duty use (OpenGL,
fast rendering) [35]"
  },
  {
    "name": "Durdraw",
    "category": "ansi-editor",
    "primary_formats": ["ANS", "TXT", "DFD"],
    "license": "BSD-3-Clause",
    "open_source": true,
    "repo_url": " [36]",
    "homepage_url": " [38]",
    "platforms": "Linux, macOS (Terminal)",
    "status": "active",
    "first_release": 2017,
    "last_release": 2023,
    "price": "free",
    "core_features": "Terminal-based text art studio; supports frame-based
animations [39]; 16-color and 256-color (xterm) modes; Unicode (UTF-8) and CP437
mixing [39]; mouse support in terminal",
    "notable_limitations":
"CLI UI (learning curve); no GUI; PNG/GIF export only in 16-color mode [40] [41]",
    "export_options": "ANS (SAUCE), JSON, DFD (internal), HTML, PNG/GIF [40]",
    "import_options": "ANS import (CP437 Unicode conversion) [39]; image→ANSI
conversion (external scripts)",
    "web_rendering_support": "Yes – HTML output, plus requires Ansilove for PNG/
GIF [40]",
    "community_metrics":
"HN discussion praised its modern terminal approach [42]; niche Unix enthusiasts
```

```
[43]",
    "docs_quality": "good (README with usage, man pages)",
    "examples_or_screenshots": "(GitHub gallery, demos available)",
    "notes": "Blends retro ANSI editing with modern terminal capabilities
(truecolor, theming) [37] [41] "
  },
  {
    "name": "TetraDraw",
    "category": "ansi-editor",
    "primary_formats": ["ANS", "TND"],
    "license": "GPL-2.0",
    "open_source": true,
    "repo_url": " [44] ",
    "homepage_url": null,
    "platforms": "Linux (ncurses)",
    "status": "archived",
    "first_release": 2003,
    "last_release": 2004,
    "price": "free",
    "core_features": "First completely usable ANSI editor for Unix [44] ; multi-
user collaborative editing over Internet [45] ; text-mode interface",
    "notable_limitations": "Unix text-mode only; no updates since early 2000s",
    "export_options": "ANS, TND, BIN, ADF",
    "import_options": "n/a",
    "web_rendering_support": "n/a (text terminal output)",
    "community_metrics": "Important historically for Linux BBS scene [44] ;
superseded by newer tools",
    "docs_quality": "okay (man page, docs included)",
    "examples_or_screenshots": null,
    "notes": "Enabled Linux users to create ANSI without DOS emulation [47] "
  },
  {
    "name": "TundraDraw",
    "category": "ansi-editor",
    "primary_formats": ["ANS", "TND"],
    "license": "GPL-2.0",
    "open_source": true,
    "repo_url": null,
    "homepage_url": " [48] ",
    "platforms": "Windows, Linux (Qt)",
    "status": "dormant",
    "first_release": 2002,
    "last_release": 2009,
    "price": "free",
    "core_features": "Qt-based GUI ANSI drawing; cross-platform; custom .TND
format for saving",
    "notable_limitations": "Beta quality (v0.5) [49] ; development halted ~2010",
    "export_options": "ANS, TND, BIN, ADF",
```

```
    "import_options": "Imports ANSI, XBIN, Artworx formats",
    "web_rendering_support": "No",
    "community_metrics":
"Allowed Win/Linux art creation in 2000s [50]; kept scene alive post-DOS [50]",
    "docs_quality": "poor (sparse readme)",
    "examples_or_screenshots": " [49]",
    "notes": "Referenced as tool that sustained late-2000s ANSI art [50]"
  },
  {
    "name": "JavE",
    "category": "ascii-editor",
    "primary_formats": ["TXT", "ASC", "XML"],
    "license": "Freeware",
    "open_source": false,
    "repo_url": null,
    "homepage_url": " [51]",
    "platforms": "Cross-platform (Java)",
    "status": "historical",
    "first_release": 2001,
    "last_release": 2013,
    "price": "free",
    "core_features": "General ASCII art editor (focus on line-drawing &
diagrams) [52]; GUI with drawing tools, text effects, image-to-ASCII converter",
    "notable_limitations":
"No color/ANSI support (monochrome only); Java app UX dated",
    "export_options": "TXT (ASCII), proprietary .jave (XML)",
    "import_options": "Image import to ASCII (bw)",
    "web_rendering_support": "Exports HTML with ASCII art",
    "community_metrics": "Once considered “the BEST AA editor” for Shift-JIS art
[53]",
    "docs_quality": "okay (bundled help)",
    "examples_or_screenshots": null,
    "notes": "Popular in early 2000s especially for Shift-JIS art; now legacy"
  },
  {
    "name": "Monodraw",
    "category": "ascii-editor",
    "primary_formats": ["TXT", "PNG", "SVG"],
    "license": "Proprietary",
    "open_source": false,
    "repo_url": null,
    "homepage_url": " [54]",
    "platforms": "macOS",
    "status": "maintained (commercial)",
    "first_release": 2015,
    "last_release": 2023,
    "price": "paid ($49.99)",
    "core_features": "Polished Mac-native ASCII art designer (diagrams,
```

```
flowcharts, layouts)[54]; shape tools, connectors, real-time table drawing;
exports to text, PNG, SVG",
    "notable_limitations": "Mac only; closed source; relatively expensive",
    "export_options": "TXT, PNG, SVG, PDF, HTML",
    "import_options": "Import ASCII text files",
    "web_rendering_support": "N/A (exports as image or HTML only)",
    "community_metrics": "Niche professional user base (developers, designers)
[55]; positive reviews on HN[56]",
    "docs_quality": "good (extensive manual)",
    "examples_or_screenshots": null,
    "notes": "Sets high bar for UX in ASCII editing; unique diagramming focus
beyond BBS art"
  },
  {
    "name": "REXPaint",
    "category": "ascii-editor",
    "primary_formats": ["XP", "TXT", "ANS", "PNG"],
    "license": "Freeware",
    "open_source": false,
    "repo_url": null,
    "homepage_url": "[59]",
    "platforms": "Windows (runs via Wine on *nix)[58]",
    "status": "maintained",
    "first_release": 2013,
    "last_release": 2023,
    "price": "free",
    "core_features": "Roguelike-centric ASCII/ANSI editor[59] with multi-layer
support[60], 24-bit color, shape drawing tools; widely used for game art/
mockups",
    "notable_limitations": "No native Linux/macOS build (Windows only, but Wine
OK)[61]; closed source",
    "export_options": "XP native format, ANS, TXT, CSV, PNG (image export)",
    "import_options": "Import XP files, plain text",
    "web_rendering_support": "Exports PNG (with transparency), ANS (16-color),
TXT",
    "community_metrics":
"Popular in roguelike dev community[62] (used in Cogmind dev); ~10k downloads",
    "docs_quality": "good (PDF manual[63])",
    "examples_or_screenshots": null,
    "notes":
"Combines retro text art and game design needs (e.g. JSON export to game
engines)[63]"
  },
  {
    "name": "ASCII Art Paint",
    "category": "unicode-art-editor",
    "primary_formats": ["TXT", "PNG"],
    "license": "GPL-3.0",
```

```
    "open_source": true,
    "repo_url": " 64 ",
    "homepage_url": " 65 ",
    "platforms": "Web (browser)",
    "status": "active",
    "first_release": 2018,
    "last_release": 2023,
    "price": "free",
    "core_features": "Web-based ASCII graphics editor 66 ; draws with
customizable symbol palettes; supports converting images to ASCII; dark & light
themes",
    "notable_limitations": "Browser-based only; limited to monochrome or simple
color (no ANSI escapes)",
    "export_options": "TXT, HTML, PNG (image of ascii)",
    "import_options": "Import images (for conversion)",
    "web_rendering_support": "Canvas element for live HTML preview",
    "community_metrics":
"Open source on GitHub (release 11) 67 ; modest adoption via itch.io",
    "docs_quality": "okay (some tutorials via itch.io)",
    "examples_or_screenshots": null,
    "notes": "Modern browser tool bridging pixel-art and text art creation 66 "
  },
  {
    "name": "ASCIIFlow",
    "category": "ascii-editor",
    "primary_formats": ["TXT", "HTML"],
    "license": "Proprietary (free)",
    "open_source": false,
    "repo_url": null,
    "homepage_url": " 68 ",
    "platforms": "Web",
    "status": "active",
    "first_release": 2011,
    "last_release": 2021,
    "price": "free",
    "core_features": "Simple online ASCII diagramming tool; grid canvas with
lines/boxes drawing; Google Drive save integration 69 ",
    "notable_limitations": "No color support; limited character set (for
diagrams only)",
    "export_options": "TXT, HTML (pre-formatted)",
    "import_options": "Import ASCII via copy-paste",
    "web_rendering_support": "Embeddable HTML (monospace text)",
    "community_metrics": "Popular for quick diagramming in code comments 70 ",
    "docs_quality": "okay (minimal help on site)",
    "examples_or_screenshots": null,
    "notes": "Lightweight, focuses on schematics/box art rather than full ANSI
art"
  },
```

```
  {
    "name": "Figlet",
    "category": "converter",
    "primary_formats": ["TXT"],
    "license": "BSD-3-Clause",
    "open_source": true,
    "repo_url": null,
    "homepage_url": " [71] ",
    "platforms": "Cross-platform (CLI)",
    "status": "active",
    "first_release": 1991,
    "last_release": 2018,
    "price": "free",
    "core_features": "Classic text-to-ASCII-art font generator [71]; dozens of
font styles (block letters, scripts, etc.)",
    "notable_limitations": "Not an interactive editor (CLI only); output is
plain ASCII (no color)",
    "export_options": "TXT (ASCII art text output)",
    "import_options": "Takes input text strings",
    "web_rendering_support": "n/a (text output only)",
    "community_metrics": "Commonly used to generate title text for ASCII art
pieces",
    "docs_quality": "good (man page)",
    "examples_or_screenshots": null,
    "notes": "Useful as plug-in for editors (e.g. generate ASCII logo text) [71] "
  },
  {
    "name": "TOIlet",
    "category": "converter",
    "primary_formats": ["TXT", "ANSI"],
    "license": "GPL-3.0",
    "open_source": true,
    "repo_url": null,
    "homepage_url": null,
    "platforms": "Cross-platform (CLI)",
    "status": "maintained",
    "first_release": 2007,
    "last_release": 2014,
    "price": "free",
    "core_features": "Enhanced Figlet-like tool that supports ANSI color output
and Unicode fonts",
    "notable_limitations": "Same limitations as Figlet: not WYSIWYG editor",
    "export_options": "TXT, ANS (colored text output)",
    "import_options": "Takes text input",
    "web_rendering_support": "n/a (console colored text)",
    "community_metrics": "Fills gap by adding color and Unicode to figlet
banners",
    "docs_quality": "okay (info pages)",
```

```
    "examples_or_screenshots": null,
    "notes": "Can be integrated into modern text art workflows for fancy
colorful text banners"
  },
  {
    "name": "boxes",
    "category": "converter",
    "primary_formats": ["TXT"],
    "license": "GPL-2.0",
    "open_source": true,
    "repo_url": null,
    "homepage_url": null,
    "platforms": "Cross-platform (CLI)",
    "status": "maintained",
    "first_release": 1999,
    "last_release": 2012,
    "price": "free",
    "core_features": "CLI tool to draw ASCII art boxes/frames around text; many
predefined styles (balloons, designs)",
    "notable_limitations": "Not an editor; limited to framing existing text",
    "export_options": "TXT (text with box outline)",
    "import_options": "Takes input text (stdin or file)",
    "web_rendering_support": "n/a",
    "community_metrics":
"Often used in combination with figlet output to create banners",
    "docs_quality": "okay (man page)",
    "examples_or_screenshots": null,
    "notes": "Could be integrated into GUI editors for one-click borders"
  },
  {
    "name": "Ansilove",
    "category": "renderer",
    "primary_formats": ["PNG", "GIF", "HTML"],
    "license": "BSD-2-Clause",
    "open_source": true,
    "repo_url": null,
    "homepage_url": " 72 ",
    "platforms": "Cross-platform (C library & CLI)",
    "status": "active",
    "first_release": 2011,
    "last_release": 2023,
    "price": "free",
    "core_features":
"ANSI/ASCII art to image converter 40  supporting ANSI, PCB, BIN, GIF animations;
retains SAUCE metadata; used by 16colo.rs for gallery images",
    "notable_limitations": "Not an editor; conversion only",
    "export_options": "PNG, GIF, HTML (from text input)",
    "import_options": "Input: .ANS, .PCB, .BIN, .ADF, .IDF etc.",
```

```
    "web_rendering_support": "Embeddable Web (Ansilove.js exists for browser)",
    "community_metrics": "Standard tool in scene for high-quality ANSI to PNG rendering [40]",
    "docs_quality": "good (docs on ansilove.org)",
    "examples_or_screenshots": null,
    "notes": "Ensures text art is viewable on web/forums without special fonts (pixel-perfect conversion) [40]"
  },
  {
    "name": "iNFekt NFO Viewer",
    "category": "renderer",
    "primary_formats": ["ANS", "NFO", "DIZ"],
    "license": "BSD-2-Clause",
    "open_source": true,
    "repo_url": null,
    "homepage_url": "(GitHub)",
    "platforms": "Windows, Linux",
    "status": "active",
    "first_release": 2010,
    "last_release": 2021,
    "price": "free",
    "core_features": "Dedicated viewer for ANSI/NFO text art with accurate colors, triple column view, zoom, and export to image",
    "notable_limitations": "View-only (no editing); Windows GUI, Linux (Qt) version available",
    "export_options": "PNG image",
    "import_options": "n/a",
    "web_rendering_support": "n/a",
    "community_metrics": "Popular for viewing scene .NFO files (releases, NFOs) with proper oldskool fonts",
    "docs_quality": "good (help file)",
    "examples_or_screenshots": "(screenshot on GitHub)",
    "notes":
"Fulfills need to read text art easily on modern OS without corrupt rendering"
  },
  {
    "name": "Teletext Editor (Edit.TF)",
    "category": "teletext-editor",
    "primary_formats": ["TTI", "TTX", "PNG"],
    "license": "MIT",
    "open_source": true,
    "repo_url": null,
    "homepage_url": " [73] ",
    "platforms": "Web",
    "status": "active",
    "first_release": 2016,
    "last_release": 2023,
    "price": "free",
```

```
    "core_features": "Browser-based Teletext art editor; authentic Mode 7 editing grid; supports level 1 teletext (24x40, 8-color) [74]",
    "notable_limitations": "Narrow use (teletext only); limited resolution and color palette",
    "export_options": "EP1, TTI, TTX (teletext page formats), PNG",
    "import_options": "Import teletext format or PNG",
    "web_rendering_support": "Canvas/WebGL (for preview)",
    "community_metrics": "Revival of teletext art community, used in teletext art comps",
    "docs_quality": "okay (tutorial available) [75]",
    "examples_or_screenshots": null,
    "notes": "Allows creation of art for teletext services and emulators; niche but active scene"
  },
  {
    "name": "PETSCII Editor (Petmate)",
    "category": "petscii-editor",
    "primary_formats": ["PETSCII", "PRG", "PNG"],
    "license": "MIT",
    "open_source": true,
    "repo_url": " [76]",
    "homepage_url": " [78]",
    "platforms": "Windows, macOS, Linux",
    "status": "active",
    "first_release": 2017,
    "last_release": 2019,
    "price": "free",
    "core_features": "Cross-platform Commodore 64 PETSCII art editor; multiple screens, undo/redo [79], custom charsets, exports to C64 formats (.prg, .asm) [77], JSON, PNG",
    "notable_limitations": "PETSCII-specific (commodore glyphs only); last update 2019",
    "export_options": "Bitmap PNG (import/export), PRG (C64 executable), ASM source, PETSCII .c format, Petmate JSON",
    "import_options": "Import .d64 disk art, .png to PETSCII (pixel-match) [80]",
    "web_rendering_support": "No (desktop only)",
    "community_metrics": "Widely used in C64 demoscene for text graphics [81]; recommended on r/c64 [81]",
    "docs_quality": "good (README, shortcuts) [82]",
    "examples_or_screenshots": null,
    "notes": "Modern tool keeping PETSCII art alive, with conveniences like multicolor mode and JSON export for integration [83]"
  },
  {
    "name": "Marq's PETSCII Editor",
    "category": "petscii-editor",
    "primary_formats": ["PRG", "PNG", "ANS"],
    "license": "GPL-2.0",
```

```
    "open_source": true,
    "repo_url": null,
    "homepage_url": " 85 ",
    "platforms": "Windows, macOS, Linux",
    "status": "archived",
    "first_release": 2011,
    "last_release": 2015,
    "price": "free",
    "core_features": "Original multi-platform PETSCII editor by Marq; supports
C64 and VIC-20 fonts, animations",
    "notable_limitations": "Superseded by Petmate (no longer updated)",
    "export_options": "PRG, standalone .exe (C64), PNG, anim GIF",
    "import_options": "Imports PETSCII prg, images",
    "web_rendering_support": "PNG export",
    "community_metrics": "Contained a PETSCII art gallery online 87 ",
    "docs_quality": "okay",
    "examples_or_screenshots": null,
    "notes": "Pioneered modern PETSCII editing; provided inspiration for newer
tools"
  },
  {
    "name": "VSXu (VSX Ultra)",
    "category": "visualizer-editor",
    "primary_formats": ["VSXu preset (.vsxu)"],
    "license": "GPL-3.0",
    "open_source": true,
    "repo_url": null,
    "homepage_url": null,
    "platforms": "Windows, Linux",
    "status": "semi-maintained",
    "first_release": 2009,
    "last_release": 2017,
    "price": "free",
    "core_features": "Standalone music visualization system with node-based
preset editor; not Milkdrop-compatible (parallel ecosystem)",
    "notable_limitations":
"Steeper learning curve (modular synth-like); smaller preset library than
Milkdrop",
    "export_options": "VSXu preset files (graph definitions)",
    "import_options": "Import images/shaders as nodes",
    "web_rendering_support": "Exports visuals via SDK (for integration)",
    "community_metrics": "Used by some VJs and demosceners as open alternative
to Winamp AVS/Milkdrop",
    "docs_quality": "okay (docs & tutorials exist)",
    "examples_or_screenshots": "(editor UI screenshot)",
    "notes": "Shows what a modern visual preset IDE can be (graphical patching
of effects), but not widely adopted outside niche"
  },
```

```
  {
    "name": "Winamp AVS (Advanced Visualization Studio)",
    "category": "visualizer-preset-editor",
    "primary_formats": ["AVS preset (.avs)"],
    "license": "Proprietary (formerly BSD-like for SDK)",
    "open_source": "partial",
    "repo_url": " [88] ",
    "homepage_url": " [89] ",
    "platforms": "Win32",
    "status": "historical (open-source community fork)",
    "first_release": 1998,
    "last_release": 2005,
    "price": "free",
    "core_features": "Classic Winamp visualization plugin featuring a built-in graphical preset editor (mix of effects, DM codes) [90]; huge library of community presets (1000s on Winamp forums, DeviantArt [91])",
    "notable_limitations": "No official updates since 2005; performance issues on modern OS; presets require old Winamp or ports",
    "export_options": ".AVS (binary preset), .APE (plugin effects)",
    "import_options": "Import user presets packages",
    "web_rendering_support": "No (real-time only; screenshots possible)",
    "community_metrics": "Thriving 2000s community (Visbot archived 15k posts [92]; AVSociety on DeviantArt) but now niche",
    "docs_quality": "okay (wiki and old forum guides) [93]",
    "examples_or_screenshots": "(AVS preset packs gallery) [94]",
    "notes": "Notable for visual programming approach to music visuals; groundwork for later shader-based systems"
  },
  {
    "name": "MilkDrop (Winamp MilkDrop 2.x)",
    "category": "visualizer-preset-editor",
    "primary_formats": ["Milk (.milk) preset"],
    "license": "BSD-2-Clause (Milkdrop 2 OSS release)",
    "open_source": "partial",
    "repo_url": " [95] ",
    "homepage_url": " [96] ",
    "platforms": "Win32 (Winamp plugin)",
    "status": "legacy (OSS code via projectM)",
    "first_release": 2001,
    "last_release": 2007,
    "price": "free",
    "core_features": "Hardware-accelerated music visualization plugin for Winamp [97]; preset-driven shader effects with audio-responsive equations; ~50k community presets created [98]",
    "notable_limitations": "Official dev halted in 2007; editing presets requires manual code tweaking or in-plugin menu [99]",
    "export_options": ".milk (preset files), Milkdrop preset pack (.zip)",
```

```
    "import_options":
"Presets can be edited live in Winamp (text UI) [99]; import old MD1 presets",
    "web_rendering_support": "Screenshots, AVI via Winamp video out",
    "community_metrics": "Iconic visualizer – still beloved; revived via
projectM & Butterchurn on modern platforms [100]",
    "docs_quality": "good (extensive authoring guides online) [101]",
    "examples_or_screenshots": "(Geisswerks tutorial, Pawel Porwisz guide) [101]",
    "notes": "Benchmarked for all modern visualizers; vibrant preset scene (73k
pack available [98])"
  },
  {
    "name": "projectM",
    "category": "visualizer-player",
    "primary_formats": ["Milk preset", "PRJM"],
    "license": "LGPL-2.1",
    "open_source": true,
    "repo_url": " [105]",
    "homepage_url": " [115]",
    "platforms": "Cross-platform (C++ library + apps)",
    "status": "active",
    "first_release": 2003,
    "last_release": 2024,
    "price": "free (GPL app/paid mobile)",
    "core_features": "Open-source reimplementation of MilkDrop [103] as a cross-
platform library and apps; runs MilkDrop presets unmodified; integrations for
VLC, Kodi, foobar2000, etc.",
    "notable_limitations": "Historically lagged behind MilkDrop on newer preset
features (now catching up); preset authoring still code-oriented",
    "export_options": ".milk, .prjm (projectM new format)",
    "import_options": "Import MilkDrop presets and playlists",
    "web_rendering_support": "Export still images, video capture in some
frontends",
    "community_metrics": "Most advanced FOSS visualizer [103]; large user base via
app stores and media players [104]",
    "docs_quality": "good (active wiki, Discord support) [105]",
    "examples_or_screenshots": "(video demos available)",
    "notes": "Ensures MilkDrop's legacy on modern systems (Win/Mac/Linux, even
Android TV) [106] [107]; constant improvements (v4+ supports modern OpenGL, JSON
preset editing, etc.)"
  },
  {
    "name": "Butterchurn",
    "category": "visualizer-player",
    "primary_formats": ["Milk preset", "JSON"],
    "license": "MIT",
    "open_source": true,
    "repo_url": " [108]",
    "homepage_url": " [109]",
```

```
    "platforms": "Web (JavaScript/WebGL2)",
    "status": "active",
    "first_release": 2018,
    "last_release": 2023,
    "price": "free",
    "core_features":
"WebGL MilkDrop 2 implementation [109] – brings Winamp visuals to browsers; high-
performance (WASM-compiled preset shaders) [110]; used in Webamp and others",
    "notable_limitations": "Limited to MilkDrop preset compatibility; no GUI
editor (presets edited as text JSON)",
    "export_options": ".milk (loaded), JSON (internal compiled form)",
    "import_options": "Milk presets (v1 & v2) load via JS, optional conversion
to JSON for speed [108]",
    "web_rendering_support": "Canvas/WebGL output embeddable in any webpage",
    "community_metrics": "Widely adopted for web music players and streams
(audio-reactive web art) [100]; active dev community on GitHub",
    "docs_quality": "good (API documentation)",
    "examples_or_screenshots": "(butterchurnviz.com demo) [109]",
    "notes": "Extends MilkDrop's reach to modern web and cross-device usage [100];
even used in Steam, Cider apps [111]"
  },
  {
    "name": "WebVS (Web AVS)",
    "category": "visualizer-player",
    "primary_formats": ["AVS preset"],
    "license": "MIT",
    "open_source": true,
    "repo_url": " [112]",
    "homepage_url": " [112]",
    "platforms": "Web (JS/WebGL)",
    "status": "inactive",
    "first_release": 2014,
    "last_release": 2015,
    "price": "free",
    "core_features": "Experimental JavaScript/WebGL reimplementation of Winamp
AVS [112]; could run many classic AVS presets in browser",
    "notable_limitations": "Project incomplete – not all AVS effects supported;
last updated 2015",
    "export_options": ".avs presets (limited support)",
    "import_options": "Import AVS presets (some may not work fully)",
    "web_rendering_support": "Canvas/WebGL output for web embed",
    "community_metrics": "Proof-of-concept level; demonstrated AVS on the web
for nostalgia",
    "docs_quality": "poor (sparse README)",
    "examples_or_screenshots": null,
    "notes": "Showed that AVS's visual logic could be ported to JS, but never
fully realized; superseded by modern shader viz"
  },
```

```
  {
    "name": "Winamp Advanced Visualization Studio DX (vis_avs_dx)",
    "category": "visualizer-player",
    "primary_formats": ["AVS preset"],
    "license": "Apache-2.0",
    "open_source": true,
    "repo_url": " [113] ",
    "homepage_url": " [113] ",
    "platforms": "Windows",
    "status": "active (community fork)",
    "first_release": 2020,
    "last_release": 2021,
    "price": "free",
    "core_features": "Updated build of classic AVS with DirectX rendering
(hardware accel) [113] and bug fixes; allows AVS to run on modern Windows and at
higher framerates",
    "notable_limitations": "No new features beyond rendering pipeline; still
Windows-only and requires Winamp/WACUP",
    "export_options": ".avs",
    "import_options": "AVS presets only",
    "web_rendering_support": "n/a (real-time only)",
    "community_metrics": "Keeps vast archive of AVS presets usable today (works
with WACUP, etc.)",
    "docs_quality": "okay (community forums)",
    "examples_or_screenshots": "(GitHub vis_avs_dx)",
    "notes": "Important maintenance update extending life of AVS visual content"
  },
  {
    "name": "AVS Plugin (MusicBee etc.)",
    "category": "visualizer-player",
    "primary_formats": ["AVS preset"],
    "license": "Proprietary",
    "open_source": false,
    "repo_url": null,
    "homepage_url": " [114] ",
    "platforms": "Windows",
    "status": "maintained",
    "first_release": 2018,
    "last_release": 2022,
    "price": "free",
    "core_features": "Wrapper allowing Winamp AVS presets in MusicBee player [114]
(and some other players); leverages community AVS builds",
    "notable_limitations": "Specific to MusicBee (closed-source plugin); relies
on legacy code",
    "export_options": ".avs",
    "import_options": "AVS presets only",
    "web_rendering_support": "n/a",
    "community_metrics": "Helps preserve old presets outside Winamp ecosystem",
```

```
    "docs_quality": "okay (MusicBee forum)",
    "examples_or_screenshots": "(MusicBee addon page)",
    "notes": "Indicative of continued interest in classic visualizations on new platforms"
  }
]
```


Notes on categories: Tools are classified by their primary purpose. For instance, **ansi-editor** vs **ascii-editor** both indicate text art editors but differ in color/format focus (CP437/ANSI vs generic ASCII/Unicode). **visualizer-preset-editor** refers to tools with editing capabilities for audio-reactive visuals, whereas **visualizer-player** is for runtime visualization engines/players.

Feature Matrix (Top Tools)

The following table compares **selected top tools** (across both ASCII/ANSI editors and visualization preset tools) against desired features. A check (✓) indicates full support, a cross (✗) for no support, and ~ for partial or limited support, often with a note. This matrix focuses on ~30 prominent tools from the catalog above:

Tool / Feature	Brush System (size/ pressure/ rotate)	Custom Pattern Brushes (tileable)	Layers & Blending	Selections & Transforms	Full Unicode Support (beyond CP437)	ANSI Color Palette & Dithering	Zoom & Large Canvas Performance
TheDraw (DOS)	✓ basic (1x1 block draw)	✗	✗	✗ (only block operations)	✗ (CP437 only)	✓ 16-col ANSI; built-in fonts [2]	~ (80x25 up to 100x100 lines; limited by DOS) [1]
PabloDraw	✓ (brush tool with sizes)	✗ (no tile brushes)	✗ (no layers)	✓ (copy/ paste between docs [116]; move)	✗ (CP437 only)	✓ 16-col ANSI + custom palette; dither options	✓ (smooth scrolling; handles large art)
Moebius	✓ (various brush shapes + half-block) [19]	~ (custom half-block patterns)	✗ (planned, not yet)	✓ (rect select, copy/ paste)	✗ (CP437 focus; Unicode planned) [117]	✓ ANSI 16-col, iCE colors; no dithering	✓ (can handle huge .ANS; GPU-accelerated)

Tool / Feature	Brush System (size/ pressure/ rotate)	Custom Pattern Brushes (tileable)	Layers & Blending	Selections & Transforms	Full Unicode Support (beyond CP437)	ANSI Color Palette & Dithering	Zoom & Large Canvas Performance
Icy Draw	✓ (brush + shape tools)	~ (supports TheDraw font “patterns”) [118]	✓ (basic layer support)	✓ (selections, move, flip)	✗ (Unicode internal but not UI yet) [117]	✓ ANSI palette + custom; dithering likely	✓ (OpenGL rendering for large canvas)
Durdraw	✗ (char-by-char only; no free brush)	✗	✗	~ (has area copy/move in terminal)	✓ (Unicode blocks, braille etc. mix [39])	✓ 16-col + 256-col xterm; dithering via custom charset	✓ (optimized for >1000 lines in terminal)
REXPaint	✓ (resizable brush, shapes)	✓ (9x9 stamps; custom tiles) [60]	✓ (layers w/ opacity)	✓ (rect select, move, rotate)	✗ (ASCII + extended CP437 only)	✓ 16-col ANSI + full RGB color picker [60] ; dithering ✓	✓ (can zoom via font scale [60] ; large maps support)
Monodraw	✓ (vector-like drawing of chars)	✗ (predefined shapes instead)	✗ (no layers, but grouping)	✓ (select, move, align, shape transform)	✓ (full Unicode charset support)	✓ Supports custom palettes and block shades	✓ (smooth zoom/pan; large canvas)
ASCII Art Paint (Web)	✓ (paintbrush tool, size adjust)	✗	✗	✓ (select, move)	✓ (Unicode symbol palette)	~ (monochrome palette; no ANSI color)	✓ (canvas pan/zoom in browser)

Tool / Feature	Brush System (size/ pressure/ rotate)	Custom Pattern Brushes (tileable)	Layers & Blending	Selections & Transforms	Full Unicode Support (beyond CP437)	ANSI Color Palette & Dithering	Zoom & Large Canvas Performance
Playscii	✓ (painting with foreground/ background)	✓ (can define custom tile set)	✗ (no layers; uses frames as separate)	✓ (selection move, copy between frames)	✓ (Unicode mode with blocks/ Braille)	✓ 256-color and truecolor modes; dither filters	✓ (handles very large canvases; OpenGL)
MilkDrop (Winamp)	✗ (no brush; code-based visuals)	✗	✗	✗ (n/a)	~ (supports Unicode characters in text overlays but rarely used)	✓ 16 million colors via shaders	✓ (efficient GPU usage even at HD)
AVS (Winamp)	✗ (visual building blocks, no brush)	✗	✗	✗	✗ (ASCII only in text render effect)	✓ 16-bit color (internally 24-bit)	~ (CPU-based, struggles at high res)
projectM	✗ (not an editor, player only)	✗	✗	✗	~ (handles Unicode in text events if preset uses; rarely)	✓ 24-bit color shader output	✓ (high performance OpenGL)
Butterchurn	✗	✗	✗	✗	~ (browser can render Unicode chars in presets if any)	✓ 24-bit (WebGL shaders)	✓ (runs 60fps in browser at 1080p)

Tool / Feature	Brush System (size/ pressure/ rotate)	Custom Pattern Brushes (tileable)	Layers & Blending	Selections & Transforms	Full Unicode Support (beyond CP437)	ANSI Color Palette & Dithering	Zoom & Large Canvas Performance
VSXu	✗ (node editor not brush-based)	✗	✗ (no layers, uses node graph)	✗	✓ (full Unicode text nodes possible)	✓ 24-bit color	✓ (GPU acceleration)
Teletext Edit.TF	✗ (cell edit only)	✗	✗	✗	✗ (Teletext charset only)	~ (8 colors + 2 levels, fixed palette)	✓ (web canvas for large pages)
Petmate (PETSCII)	✓ (char draw, brush select)	✗ (no tile brush, but char selection)	✗ (no layers, multiple "screens" instead)	✓ (select, shift screen) [119]	✗ (PETSCII only)	✓ Commodore 16-color + multicolor	✓ (smooth canvas pan/ zoom)
Winamp MilkDrop Preset Editor	✗ (no GUI brush; code only)	✗	✗	✗	~ (can render Unicode if coded)	✓ (custom palette via shader)	✓ (fast GPU visuals)

Key observations from the matrix:

- **Advanced drawing features:** Modern text editors like **REXPaint** and **Moebius** offer *brushes, shapes,* and even **half-block painting** (Moebius) for finer detail [19] . **Layers** remain relatively rare – *REXPaint* and upcoming tools like *Icy Draw* support layers, addressing a frequent request for complex compositions (users on Reddit were excited about layers in Icy Draw). Selection and transformation (move/rotate/flip) are common in GUI editors (PabloDraw, Moebius, RexPaint), though absent in terminal-based ones.

- **Character set and Unicode support:** Older tools are tied to code page 437, whereas newer ones see **demand for Unicode** (e.g. *Durdraw* mixing CP437 with Unicode braille blocks [39] , and a user wishlist explicitly called out Unicode glyph support [120]). **MoebiusXBIN** takes this further by allowing custom fonts/encodings, effectively any Unicode symbol set [22] . This is a clear trend: modern ASCII art apps must go beyond CP437 to include box-drawing, block elements, and even emoji.

- **Palette and color:** All ANSI editors handle the classic 16 colors; many now also handle extended palettes (256-color ANSI, truecolor). For instance, Durdraw supports both 16-color ANSI and 256-

color xterm mode [39], and RexPaint even has a full RGB picker [60]. Dithering tools (for converting images to ASCII) exist externally and sometimes internally (Playscii has some dithering filters). Palette management (loading C64 palettes, custom ANSI palettes, etc.) is increasingly important (MoebiusXBIN includes many palettes [22]).

- **Canvas size and performance:** New editors are built to handle **very large text canvases** (hundreds of lines/columns): Moebius and Icy Draw leverage GPU acceleration (OpenGL) to scroll/zoom smoothly, overcoming old DOS limits [4]. Playscii and RexPaint also handle large images with ease. This addresses the historical 100-line barrier in TheDraw [4] – modern artists expect to create much bigger pieces (e.g., for posters or animated ANSI).

- **Collaboration & version control:** PabloDraw pioneered multi-user editing over TCP/IP (multi-cursor in the same canvas) [12], which remains unique. Most other tools are single-user. However, with ASCII art often being a niche, real-time collab isn't common, though it's a potential differentiator for a new app (especially given the success of collaborative coding and design tools).

- **Animation support: Frame-based animation** in text art is now a reality in tools like Durdraw [39] and Playscii (which can onion-skin between frames for ASCII GIF creation). This resurrects the "ANSiMation" concept in a modern way. Expect demand for this as ASCII art is used in creative videos or terminal animations.

- **Automation and scripting:** Very few text art editors support scripting – Playscii has a Python API, and some command-line tools like FIGlet/boxes can be scripted in pipelines. This is an area of opportunity (e.g., batch-processing images to ASCII, algorithmic art generation).

- **Export and integration:** Modern ASCII/ANSI editors typically provide **image export (PNG)** for sharing on the web (since not all platforms can render .ANS correctly) – often via Ansilove or internal routines. HTML export (with CSS for colors) is also seen (Durdraw, ASCII Art Paint). **Embeddable viewers**: some editors (PabloDraw, sixteencolors.net) rely on web libraries to display art – a strong candidate for a new project is a built-in web viewer (perhaps leveraging canvas or WebGL for perfect fidelity, like Ansilove.js).

- **Accessibility:** Text art inherently can be accessible (as it's text), but the editors themselves need considerations (keyboard navigation, high contrast UI, screen reader labels). Most older tools are keyboard-driven (by necessity on DOS/terminal), which is good, but UIs were not designed with screen readers in mind. A modern editor could capitalize on this by ensuring UI controls are accessible and perhaps offering assistive features (like describing ANSI art out loud – a challenge but notable).

- **Visualizer tools special notes:** For Winamp-style visualizers, the "features" differ. The **preset editors** (MilkDrop, AVS) are more like IDEs for generative graphics:

- MilkDrop 2's "editor" is essentially a real-time code editor for shader-like scripts [99], so no brushes or GUI aids – a gap noted by users (authoring MilkDrop presets has a high learning curve [101]). Some have dreamed of node-based or GUI tools for MilkDrop presets; none have fully materialized, though projectM's team has discussed GUI editors (and one can imagine integrating a Shadertoy-like interface).

- AVS had a GUI for selecting effect components, but it's dated. **No modern visual preset creation tool exists with a friendly UI**, representing an opportunity. Community interest is evidenced by attempts like WebVS and forum discussions on making preset creation easier [93].

Audio-reactivity is a given in these, but **ease of use and cross-platform playback** are what recent projects address (projectM and Butterchurn enabling use outside Winamp). A new tool focusing on visualizer preset editing could combine the intuitiveness of AVS's building blocks with the power of MilkDrop's shaders.

In summary, the feature matrix shows that **no single tool yet ticks all the boxes**. For ASCII/ANSI art, an ideal modern editor would combine: multi-layer compositing, high-zoom canvas, Unicode + custom fonts, integrated figlet/boxes, animation support, and collaboration – none of which are all in one product (Icy Draw and Moebius are getting close, but still evolving). In the visualization realm, there's a clear gap for a cross-platform preset *editor* (not just player) that could lower the bar for creating those mesmerizing graphics.

Market Brief

Landscape Summary

Text Mode Art Revival: ASCII/ANSI art, once a necessity of BBS culture, has become a **retro-nostalgic art form** sustained by passionate communities. Platforms like **16colo.rs** (an archive for new and old ANSI/ASCII art) and events at demoparties (text mode graphics compos) show continued interest. The scene is vibrant albeit niche: - **Communities & Groups:** Long-standing art groups (ACiD, iCE) wound down in early 2000s [121], but new collectives like **Blocktronics** (founded 2008) carry the torch [122]. There's cross-pollination with the **demoscene** (many ANSI artists are demosceners and vice versa). The **Mistigris** group and Discord servers (e.g., "Artscene" Discord) host active discussions. Subreddits like r/ANSIart and r/bbs are small but active, indicating a resurgence of interest among tech nostalgists and young artists experimenting with the style. - **Tools Maturity:** We now have both **historical tools** (TheDraw, ACiDDraw – useful for inspiration but only run in DOS/emulators) and **modern ones** (PabloDraw, Moebius, etc.). Modern tools are mostly freeware or open-source, often labors of love by lone developers or small teams. A few commercial exceptions exist (Monodraw on Mac, primarily targeting diagramming rather than ANSI art enthusiasts). Many open-source projects see sporadic development, often when the maintainer is personally interested (e.g., Moebius by Andy Herbert with last updates in 2023, Durdraw by cmang still active, Icy Draw just emerged in 2023). - **Cross-platform expectation:** Unlike in the '90s (when ANSI art was mostly made on DOS), today's users demand editors for Windows, macOS, and Linux alike. Cross-platform frameworks (Qt, .NET 5/6, Electron) have enabled this – PabloDraw leveraged .NET Core, Moebius uses C++/SDL, etc. **Web-based editors** are also gaining traction for accessibility (no install needed) – e.g., ASCII Art Paint, ASCIIFlow for quick diagrams, and various image-to-ASCII converters online.

Usage & Users: ASCII/ANSI art creation now falls into a few niches: - **Art packs and galleries:** Enthusiasts still release periodic "artpacks" (often in .zip files containing .ANS, .NFO, .TXT with SAUCE metadata) reminiscent of the BBS era. These packs are curated on 16colo.rs and similar sites. Newcomers often discover ANSI art through these archives or via social media, then seek tools to try it themselves. - **Roguelike and game dev:** Ascii art editors like RexPaint have found a secondary market in indie game development (for prototyping roguelike graphics, designing levels with text symbols, etc.). This increases demand for features like multi-layer, high resolution, custom tilesets, which pure BBS ANSI editors historically didn't need. - **Developers/DevOps:** A subset uses ASCII diagramming tools (e.g., Monodraw,

ASCIIFlow) for technical documentation, because text diagrams fit version control and code reviews well. This is adjacent to the ANSI art scene but has different requirements (monochrome, box drawing characters, etc.). It shows a market for "text graphics" beyond just art – any modern tool that can straddle diagramming and art (with modes or templates for each use-case) could capture both audiences.

What's Missing / Gaps: Despite the variety of tools, certain gaps are evident: - **Unified Feature Set:** Many tools excel in some areas and lack others. For example, one might have networking collab but no Unicode support, another has Unicode but no GUI. There isn't yet a *one-stop editor* that ticks all the requested features the community has voiced (like the list of "must-haves" on Reddit: Unicode, 256-color, layers, cross-platform terminal mode [120]). - **Ease of Use vs. Capability:** Some modern editors (e.g. Moebius) aim to mimic the feel of pro design software (Photoshop-like), which can be complex for newcomers. Conversely, old tools are simple but limited. There's room for a tool with **progressive complexity** – easy to start drawing ASCII, but with advanced panels for layers, scripting, etc. as you need them. - **Integration with Modern Workflow:** The ability to easily share creations on modern platforms is crucial. That means export to web-friendly formats (SVG for crisp vector-text, PNG for bitmap snapshot, even animated GIF/video for ANSI animations). Not all tools handle that well yet (some rely on external converters). A modern editor bundling a high-quality exporter (like a built-in Ansilove engine or similar) would save users hassle. Also, developers might want to generate ASCII art as part of build processes (for readme files or console output graphics) – a scriptable CLI for the editor or library form could tap into that demand.

Ecosystem & Formats: Alongside editors, there are ancillary components: - **Fonts & Encoding Resources:** Websites like **int10h.org** (VileR's archive of text mode fonts) are important for reference – artists want their art to look right with the proper fonts. Tools that incorporate these (like MoebiusXBIN including VileR's fonts [123], or Icy Draw supporting .TDF TheDraw fonts) are appreciated. - **Metadata (SAUCE):** Preservation of SAUCE tags (which store title, author, group, date, etc.) is considered good practice in the scene. Modern editors all try to support it. It's a selling point if your tool properly reads/writes SAUCE – avoids the nightmare of someone losing attribution info after editing in a tool that ignores it. - **Conversion & Viewing:** Tools like Ansilove, and viewer apps like iNFekt, are part of the pipeline. While not "creative" tools, they ensure art can reach audiences. Any new editor would ideally integrate or bundle such viewers or converters to simplify the user's workflow (e.g., a one-click "export as PNG" or a live HTML preview panel for the ANSI art).

Audio Visualization Ecosystem (Winamp-style Visuals): - The "preset" communities peaked in the Winamp era (2000s) with massive libraries of presets shared via forums like Winamp Forums (MilkDrop forum threads still exist) and DeviantArt (e.g., AVS presets in groups like AVSociety [91]). After Winamp's decline, these communities shrank, but **notable fragments remain**: - **projectM** has a community of open-source enthusiasts (as seen on their GitHub and Discord) trying to keep these visuals alive on new music players. - **Visbot** is a group dedicated to archiving and continuing AVS preset creation (they even put AVS 2.82 source on GitHub and made minor updates). - A niche of electronic music streamers and VJs still use MilkDrop (or projectM) for live visuals. For example, tools like **Kaleidosync** (a MilkDrop-based VJ tool) existed, and Butterchurn integration in online radio or DJ apps is happening. - There's a **new interest in using these visuals in creative coding**: e.g., the HuggingFace MilkDrop LLM mentioned in links suggests experimentation with AI-generated presets – a sign that people still find the concept intriguing and worth exploring with modern tech.

- **Modern Relevance:** The resurgence of Winamp (the brand is relaunching) and projects like Webamp (a full Winamp in the browser) bring nostalgia visuals to new audiences. Butterchurn has made it

trivial to embed MilkDrop in any web page; we're seeing it used as a "cool background" on websites, in music web players, etc. So MilkDrop presets are finding a second life on the web. However, AVS presets (which are more CPU heavy and Windows-tied) haven't enjoyed such a revival – they remain more of an underground, preserved art form.

- **Opportunity:** No one has yet built the "Figma for music visualizers" – a modern app to graphically create and tweak these audio-reactive visuals, share them, maybe even interface with music streaming. Given how popular visualizers are in video content (think Spotify's Canvas feature or YouTube audio-reactive visuals), there's potential to modernize the preset concept to a broader creative tool (beyond Winamp). MilkDrop's tech (per-pixel shaders reacting to FFT of music) was ahead of its time. Today, with WebGL, shaders are mainstream (e.g., Shadertoy community). A modern UI on top of something like Butterchurn could attract a new generation to preset-making, framing it almost like a game or creative coding platform.

Social & Nostalgia Trends: - We're firmly in an era where the aesthetics of the 80s/90s (pixel art, VHS glitches, text-mode graphics) are *retro cool*. ANSI art fits into this trend (for example, we see ANSI-style illustrations in zines, album covers, even fashion). The demoscene has "Oldskool" textmode competitions, and events like **Blockparty** had ANSI/ASCII compos. Teletext art had a mini-revival (e.g., art shows displaying teletext pages on old TVs). These are small scale but indicate a general cultural appreciation. - **Streaming & Video** – people streaming themselves coding or gaming sometimes use ASCII art generators to decorate their streams. The Matrix "digital rain" (effectively green text art) is a pop culture icon. All this to say, text art has a mainstream appeal when packaged right (see how fast an "ASCII filter" can go viral on social media). A modern tool might succeed by not just targeting the old BBS crowd, but also new users who are after the *aesthetic* for their creative projects.

Competitive Highlights: - **Open Source vs Commercial:** In ASCII/ANSI editors, open-source/free reigns – the target user base expects tools to be free (and many are hobby projects anyway). Commercial attempts like Monodraw are rare and succeed by targeting a different market (macOS power users, diagramming) rather than the art scene directly. Similarly, in visualizers, projectM and Butterchurn (open) have effectively superseded any need for a paid product (the original Winamp MilkDrop was free with Winamp). One exception: some mobile visualizer apps based on projectM sell for a few dollars on app stores – indicating casual users might pay a bit for convenience on their phone/TV. - **Quality and Continuity:** A lot of current tools are *maintained but not heavily developed*. This means a new entrant with robust features could capture attention, especially if the developer engages with the community for feedback (the Icy Draw dev asking Reddit for input is a great example [124] – it built goodwill and ensured features matched real needs like the BBS line-length option [125]). Being responsive to such niche needs (e.g., supporting PCB @X codes for BBS, or special PETSCII quirks) can quickly make a tool the favored one among enthusiasts. - **Integration with Platforms:** One competitive edge could be integrating with modern platforms – e.g., a text art editor that can post directly to Twitter with proper formatting, or a visualizer tool that can ingest Spotify or microphone input easily and output to OBS for streamers. Those are outside the scope of "traditional" tools but align with how people might want to use them today.

In summary, **the market is niche but global** – pockets of users across the US, Europe (especially Scandinavia for demoscene, Eastern Europe for teletext, etc.), and pockets in South America and Asia (Shift-JIS art in Japan remains a thing on textboards). A modern cross-platform editor that is open-source (or affordable) and embraces this culture could become the central tool of a modest-sized but passionate

market. It's not about huge revenue, but about *dominating the niche* and possibly spilling over to adjacent niches (like diagramming or VJing visuals), as the user mentioned.

Trends and Opportunities

- **Web-based Creation and Sharing:** As seen with web apps like ASCII Art Paint and the popularity of Butterchurn's website, lowering the friction to create and share is key. An app that runs in the browser (or has a companion web app) can attract casual users. Imagine a web gallery where people can draw ANSI art collaboratively or a site where users remix MilkDrop presets visually – these could spur a renaissance in content creation. There is an opportunity to integrate creation tools with **online galleries/communities** (for example, a direct export from an editor to 16colo.rs, or an online preset repository).
- **Cross-pollination between ASCII art and code art:** The rise of creative coding (Processing, p5.js, Shadertoy) shows people love to create visually with code. ASCII art and visualizer presets both sit at an intersection of programming and art. Marketing a modern tool as a creative coding platform (with a gentler learning curve) could attract users from that sphere. E.g., include a scripting engine (Lua or Python) to manipulate the ASCII canvas or generate patterns algorithmically – now your tool is also appealing to generative artists.
- **Social media formats:** Perhaps support exporting ANSI art as animated SVG or even as an HTML5 canvas snippet that can be embedded. For visualizers, providing templates to easily make music videos (say, import an audio file and auto-generate a video with chosen preset) could open a use-case for indie musicians who want cool visuals without mastering AfterEffects.
- **Educational angle:** These tools also teach programming concepts (MilkDrop presets teach about shaders and math; ASCII art can teach thinking in grids). There might be a minor opportunity in educational tech – e.g., a simplified visualizer editor used in a digital art class to demonstrate sound-reactive programming, or an ASCII art tool to teach image processing (when converting pictures to ASCII). While not a primary market, it's a way to broaden appeal.

Competitors Overview: - In ASCII art editors: The main "competitors" to a new project are Moebius (open, actively improved), PabloDraw (stable, widely used but not evolving much), and newcomers like Icy Draw. A new entrant would need to outdo them in features or UX. Given Moebius's strong foundation, one strategy could be to build on it (it's open source Apache2) – e.g., contribute features or fork if needed (like MoebiusXBIN did). The community often consolidates around a few key tools to avoid fragmenting too much. If your features are unique (say, real-time collaboration or a built-in web publishing), that could differentiate strongly. - In visualizers: Right now, projectM (with its Qt and SDL frontends) and Butterchurn (web) dominate usage. AVS is legacy/hobby only. A "competitor" could be something like **Magic Music Visuals** (commercial software for custom music visuals, node-based, not MilkDrop-related). Magic is powerful and used by some VJs, but it's $40+ and not focused on the MilkDrop aesthetic. If a new tool combined MilkDrop's library of presets with a friendly UI, it could undercut commercial tools for a segment of users who specifically love that retro look. Additionally, as VR and AR come up, music visualization might find new contexts (imagine an AR text rain or VR ANSI tunnel – speculative, but the tech could be repurposed).

Gaps/Opportunities Mapped to Desired Features

1. **All-in-one Modern Text Art Suite:** There is a clear opening for a **cross-platform ANSI/ASCII editor** that combines the best of all predecessors:
2. Full CP437 and Unicode support,

3. Advanced drawing tools (yes, brushes and shapes in a text context),
4. Multi-layer compositing with alpha (for complex ANSI overlays or blending ASCII with background images),
5. Animation timeline for ANSI (something only Durdraw attempted in terminals),
6. Integrated figlet/boxes for ASCII banners,
7. Collaboration (even if not heavily demanded, it could spur new community interactions like live multi-user art jams).

No current tool does all this. The market size might be small, but being **the tool** every ASCII/ANSI artist switches to is achievable by hitting these notes. Monetization could be via donations or a modest price on app stores (but given precedent, open-source is the more accepted route here).

1. **Web Integration & Preservation:** Offering web publishing (one-click upload to a gallery with proper credits and SAUCE preserved) could leverage the nostalgia on social media. Also, ensuring output works on modern terminals (e.g., testing art in Windows Terminal, VS Code, etc.) and providing an embeddable viewer (like a script tag that can render ANSI on a webpage accurately) could make the tool not just an editor but a platform for experiencing text art.

2. **Revamped Visualizer Authoring:** For Winamp-style visualizations, the creative community has dwindled partly because the tools to create presets remained arcane. A modern UI (maybe node-based like AVS but using modern shader building blocks) could attract both old preset authors and newbies who are intimidated by code. This tool could coexist with the editors or be a mode within a larger creative app. An example concept: a unified "Demoscene Lab" where one can draw ASCII art *and* craft music visualizers, since both are demoscene-adjacent arts. This is an ambitious merge of domains, but it could set a product apart.

3. If not combined, at least learning from each when designing: e.g., shader visualizer could borrow timeline ideas from animation editors; ASCII editor could borrow audio-reactivity for fun (why not allow animating ANSI art to music? Some BBS intros did scroll text to music beats).

4. Given MilkDrop presets are now basically GLSL shaders, tapping into the vibrant **Shadertoy** community is an opportunity: maybe allow import of simple Shadertoy shaders or provide a library of common visual effects so users can "tweak a preset" with sliders instead of coding. There's definitely interest – people still share new MilkDrop presets in the Winamp forum up to 2020s, and projects like the MilkDrop Neural Network (HuggingFace) hint at a desire to automate or simplify preset creation.

5. **Mobile & Tablet Experiences:** No major ASCII art editor exists on Android/iPad (except some barebones apps). Tablet with pencil could be awesome for drawing text art (hand-drawn ANSI? might be neat). On the visualization side, phones and tablets are powerful enough to do these effects (projectM is on Android). A good touch UI for either drawing ASCII or performing visuals live (touch VJ controls) could open a new user segment (e.g., hobbyist VJs, or just bored commuters making ASCII doodles). If resource and scope allow, having a mobile companion app – even read-only (viewer) or simple editor – enhances the ecosystem.

Competitive Highlights

Strengths of incumbents: - *PabloDraw:* rock-solid, widely compatible, multi-user sync – but stagnated in features. - *Moebius:* cutting-edge features (half-block, cross-platform, open) – but not 1.0 yet, no animations or collab. - *Icy Draw:* new and actively soliciting features – could evolve fast; currently has momentum. - *projectM/Butterchurn:* have solved cross-platform playback – any new preset tool can build on their playback engines rather than reinventing. - *AVS (old) and others:* tons of legacy content to leverage (if you can support/import it, you instantly have a library of effects for users to study or remix).

Weaknesses/gaps: - Many tools have poor UI/UX by modern standards (e.g., steep learning curve, or terminal UI). This scares off newcomers who might otherwise dabble in ANSI art. A modern app with a friendly UI (tooltips, modern design conventions) could win adoption from curious new users who stumble on ANSI art. - Lack of documentation is common in open-source projects; a project that provides good docs, tutorials, and maybe community plugins (think an extension system for new effects or formats) would stand out.

Uniqueness & Ecosystem Leverage: - If the new tool can leverage existing ecosystems – e.g., allow plugging in **FIGlet fonts, TheDraw .TDF fonts, ImGUI for scripting GUI, Shadertoy importer** – it gains functionality by standing on shoulders of giants. - Integration with **Mystic BBS or other modern BBS software**: Mystic BBS (a modern BBS package) has a built-in editor; maybe a modern external editor could interface directly with a BBS to upload screens or allow "live editing" of a BBS menu. That would be novel. - Possibly connect ASCII art with the **NFT/art blockchain trend** (there was a brief fad of generative text art NFTs). While speculative, supporting outputs like SVG (vector text art) could allow people to print high-res posters of their ANSI art or sell them as digital collectibles. (There was an instance of an ANSI art piece sold as NFT by an artist, if memory serves.)

Overall, the market brief identifies that while these domains are somewhat retro-niche, they are experiencing a renaissance powered by nostalgia and new technology (web, open source, demoscene events). A well-crafted modern tool, if properly communicated, could become the go-to platform in these circles. The key is to respect the rich legacy (support old formats, honor the culture of attribution and community) while injecting fresh ideas and convenience that today's users expect.

Ranked Feature Wishlist (MVP → V1 → V2)

Finally, based on the research, here is a prioritized wishlist of features for a hypothetical **modern cross-platform text-mode art and visualization editor**. The features are ranked by a combination of user demand (as evidenced by community discussions), uniqueness (how much it sets the tool apart), development effort (feasibility), and how well it leverages existing ecosystems:

MVP (Minimum Viable Product) – Core essentials to satisfy basic needs:

1. **Core ANSI/ASCII Editing Tools:** A robust drawing interface with **resizable brush** (single char and "stamp" modes), **color palette picker**, line/rectangle/ellipse tools, fill tool. These are table stakes – virtually all artists expect them (PabloDraw, Moebius have these) and lack of these is a non-starter.
2. **CP437 and Unicode Support:** Ability to switch between classic 256-character mode and a Unicode mode. As user *XaviousD* put it, "Unicode glyphs" and "256 color XTerm" support open new possibilities [120] . This is high demand especially for new-school artists who want to use block

elements, braille patterns, or mix scripts. *(Evidence: Reddit comments listing Unicode as #1 feature [120]; Durdraw and MoebiusXBIN making it a priority.)*

3. **Multi-platform Release:** From day one, have builds for Windows, macOS, and Linux. The community is diverse and being inclusive gains goodwill. Tools like Moebius and Icy Draw earned praise by being cross-platform out of the gate [30]. Use a cross-platform framework (Qt, Electron, .NET MAUI, etc.).
4. **Import/Export Formats: ANS, PNG, and SAUCE** support are critical. The tool must read/write ANSI (.ANS) with SAUCE without data loss (or people will avoid it) [40]. PNG export (for sharing on web easily) should be built-in (leveraging something like Ansilove library internally to ensure accuracy [40]). Export to plain TXT and HTML (for ASCII art) also high value. *(Evidence: Every modern tool touts its export options, and users often ask "can it export to PNG?" on forums – e.g., Durdraw instructs installing Ansilove for PNG export [40].)*
5. **Undo/Redo & Auto-save:** A deep undo stack (not just one-level like TheDraw) is expected now. Autosave or recovery is also important – losing hours of art to a crash is painful (older tools were not always stable). Modern UX expectations.
6. **Basic UI Modernity:** Things like zoom (people want to zoom in to do pixel-precise placements) – e.g., RexPaint scales by changing font size on the fly [60]. Smooth panning, grid toggles, perhaps optional guidelines for text baseline or centering. These might seem minor, but they drastically improve usability and match modern design tool feel.

(MVP rationale: These features ensure the tool is actually usable and not missing fundamentals that every other tool has. If any of these were missing, it would be hard to get users to switch.)

V1 (First full version) – The standout features that address unmet needs:

1. **Layers and Blending:** Implement multiple layers of text with blending modes (at least simple alpha or additive). **Demand:** High – artists have long wanted to compose complex scenes in parts (one Redditor explicitly listed "5: Layers" as something they want to see in editors [126]). RexPaint's popularity in part comes from its multi-layer ability [60]. Layers would be relatively unique in ANSI art (only RexPaint and Icy Draw have it in some form).
2. **Selections & Transformations:** Advanced selection tools – not just rectangular, but lasso or flood (magic wand by color) selection if possible, and the ability to move, rotate (90-degree steps at least), flip horizontally/vertically the selected region. **Evidence:** Users mentioned using two instances of editors to copy/paste between files as a workaround [116] – a single app that lets you copy-paste across canvases or files would save time. Transformation requests come up as well (even if rotation in text can be lossy except 90°, it's useful for box-drawing art).
3. **Integrated FIGlet/Fonts and Text Generators:** Include a library of fonts (TheDraw's .TDF fonts, FIGlet fonts, etc.) and a tool to type a string and insert an ASCII art banner. **Demand:** Implied by how almost every artist at some point uses FIGlet or TheDraw's font feature for titles. Icy Draw's dev was asked to include TheDraw fonts [33]. Having this built-in would draw users who currently rely on external tools and then import.
4. **Pattern Brushes / Stamps:** Ability to select an area (e.g., a 2x2 or 3x3 block of characters) and use it as a repeating brush. Great for drawing borders, textures, etc. **Evidence:** Historically, artists create "blocks" manually or copy-paste repeatedly. Moebius's half-block brush is a form of this [19]. This feature increases efficiency for common patterns (like checkerboard, shading blocks). There's clear value, though not widely advertised by users (because many don't imagine it yet).
5. **256-color / Truecolor mode:** For Unicode/UTF-8 art, allow use of 256-color extended ANSI or even truecolor (24-bit) in text (supported by modern terminals). This intersects with "demoscene new school" where ASCII can have many colors (e.g., Blinkenshell ASCII with extended palette). Durdraw

shows there is interest in 256-color text art [39]. It's a unique selling point, even if not everyone uses it, it future-proofs the tool.

6. **Extend to PETSCII/Teletext Modes:** Possibly as modes or plugins – allow switching canvas to PETSCII character set or Teletext (size 40x25 with constrained color rules). This taps into those sub-communities. E.g., Petmate is great but if your tool can do both ANSI and PETSCII by just switching mode, multi-talented artists will prefer one tool. Given open-source code exists to handle those (Petmate, edit.tf), integration or inspiration is feasible.
7. **Embedded Media Export:** For example, export an ANSI drawing as an HTML5 canvas animation or an SVG (with each character as a <text> element) – the latter is super useful for making high-resolution prints of ASCII art or scalable designs. Monodraw's ability to export SVG is appreciated by diagrammers [56], and ASCII artists have used hacks to get vector output for poster printing. Including this would be unique in the ANSI space.
8. **Preset Visualization Editor (MilkDrop/AVS):** Introduce a separate mode or companion app for visualizers. For v1, maybe start simple: a **MilkDrop preset tweaker** – GUI sliders to modify common variables (wave amplitude, colors, etc.) and a text editor with syntax highlighting for the preset equations. Over time, evolve this into a more visual node editor. This feature crosses into the second domain of the research. It's a bigger project on its own, but even basic functionality (like a UI to browse a folder of presets, preview them with your music, and edit values) would delight the small but passionate audio-visualizer crowd. **Demand:** Niche but high among those who care – the absence of a good editor is often lamented in forums (e.g., "is there a better way to make MilkDrop presets?"). Given our combined app concept, this is a unique selling point (no other ASCII art tool has this; it'd stand out as a demoscene multi-tool).

(V1 rationale: These features set the tool apart from any existing single competitor. They directly address many "I wish I had" moments from users (layers, better font tools, extended colors) and leverage other ecosystems (fonts library, extended char sets). They would likely make the tool the most feature-rich in its class.)

V2 (Next versions / Stretch Goals) – Innovative or heavy-lift features:

1. **Real-time Collaboration:** Multi-user editing on the same canvas over internet (like Google Docs but for ANSI art). PabloDraw did this on a LAN/Direct IP basis [12]; doing it with modern networking (maybe using web tech or cloud) could revive the fun of group draws. It's technically complex (synchronization, conflict resolution), hence slated for later. But it's a unique feature that could attract people for the novelty and community events (collaborative art jams).
2. **Plugin/Script System:** Allow advanced users to extend the tool – e.g., a Python or Lua scripting console to manipulate the canvas (for procedural effects like "wave warping" the text, or automatically importing an image and dithering it). This taps creative coders and could lead to user-contributed plugins (much like GIMP or Photoshop have). Visualizer side: maybe allow plugins to add new waveform analyzers or effects.
3. **Audio-Reactive ANSI Mode:** Here's a wild idea: merge the two domains – let the editor have an **"audio reactive" mode where ANSI art can respond to music (as a visualization)**. For example, you could place some special markers or use certain characters that flip between two states on beat, etc. Essentially an ANSI demo maker. This would be totally novel – turning static text art into a simple music visualization. It could export as a series of ANSImation frames or a video. It leverages our visualizer code but applies it to text art. This could result in fun outputs (imagine an ASCII equalizer that jumps in your ANSI image). This is definitely a niche within a niche, but it would embody the demoscene spirit and generate buzz.

4. **Android/iPad App Variant:** Depending on tech stack, either a full port or a companion viewer. The reason it's valuable: show your ANSI art on your phone, or sketch ideas on the go. For visualizers, an Android app is essentially already done by projectM for playback, but an editor on tablet for presets or text art (with pencil support on iPad for hand-drawn text strokes) could be groundbreaking. This has moderate demand; for instance, an artist might want to draw on a tablet with a stylus instead of mouse – currently impossible for ANSI art. It's a lot of dev work though, hence V2.
5. **JS/Web Embeddable Viewer Library:** By V2, formalize the viewing side: a JavaScript library that can take an ANSI/ASCII file (with SAUCE) and render it exactly as the editor would, in a web canvas or DOM. This can be offered as a standalone library (like "TextArt.js"). It leverages internal rendering code. Demand is there – many sites roll their own (16colo.rs uses Ansilove web, etc.), but an officially supported viewer means any art shared can include a mini viewer ensuring it looks right. It also promotes the editor (free marketing, as art posted with that viewer could say "made with X"). Since the research specifically noted web rendering libs [72], providing one consolidates ecosystem around your tool.
6. **Performance Benchmark: 100K characters and Beyond:** Ensure that by V2 the app can handle extremely large artworks or animations (for instance, an ASCII art of size 1000x1000, or a loop of 200 frames) without crashing. This might involve optimizing data structures (maybe using tilemaps or GPU acceleration for rendering). It's more of a technical goal than a feature, but it's something that will allow new kinds of usage (like giant banner prints or long scrolling animations).
7. **Integration with BBS/Network:** Possibly a feature to **connect to a live BBS** to edit its menus online, or at least an uploader to BBS (via telnet/SSH) to test how your ANSI looks on an actual board. Since some BBSes still run (and new ones on telnet), this would delight sysops. Not mass-market, but it fits the ethos of catering deeply to the scene. (One user in the Reddit thread was asking about BBS-specific features like @ codes for user name insertion [127] – our tool could include templates for major BBS software codes, making life easier for BBS modders.)

(V2 rationale: These are the "wow" features that could either push the envelope of what's possible or integrate the tool deeply into workflows. They're higher effort or more experimental, which is why they come later. They could also attract attention beyond the immediate community – e.g., collaborative ASCII art might get tech blog write-ups; audio-reactive ASCII might create a new micro-genre of art.)

Evidence & Demand Recap for Wishlist: - *Layers & Unicode (V1)* – explicitly requested by multiple community members [120] [126]; proven useful in RexPaint, Icy Draw. - *Brushes & Patterns (MVP/V1)* – Moebius's success with half-block brush [19] shows innovative brush tech is appreciated. Figlet inclusion frequently comes up in tool discussions. - *Collaboration (V2)* – PabloDraw's enduring use for network sessions indicates some continued interest, though it's niche. It's the kind of feature that isn't demanded loudly but when available could spark new usage (surprise-and-delight). - *Web integration (V2)* – The research noted multiple ANSI-to-web projects, meaning developers have spent effort here [72]. A unified solution from the editor side would meet a known need. - *Audio visualization editing (V1/V2)* – While not a mass request (because few imagine it possible beyond Winamp), the enduring MilkDrop preset community and projects like new preset packs [98] and Butterchurn show the content still has value. A user-friendly creation tool is the logical next step; given no one else has done it, even moderate demand is unmet.

In conclusion, the roadmap prioritizes getting the basics perfect (so the existing user base can comfortably migrate to the new tool), then introducing the most asked-for enhancements (layers, Unicode, extended colors), and finally innovating in ways that merge the ASCII art and visualization domains or otherwise leverage modern tech (collab, scripting, AR/VR possibly). By following this wishlist, the resulting product would not only cover the current **market gaps** identified (lack of a one-stop modern editor, and lack of a

modern preset editor) but also push the boundaries of what's considered text-mode art and visualization, potentially growing the market by inspiring new creative use-cases.

---

1 4 5 6 7 8 9 10 11 DarkDraw (text art editor for DOS)
https://saul.pw/darkdraw/

2 3 50 121 122 ANSI art - Wikipedia
https://en.wikipedia.org/wiki/ANSI_art

12 PabloDraw - Facebook
https://www.facebook.com/p/PabloDraw-100064543125857/

13 72 88 89 90 91 92 93 94 98 112 113 114 links.md
file://file-61xuZbhvYbtxYgmBh1Rv6u

14 Amiga ASCII art - GlyphDrawing.Club
https://blog.glyphdrawing.club/amiga-ascii-art/

15 PabloDraw (Win) 3.2.1 - SynchroFans
https://tcb.synchronetbbs.org/index.php/downloads/download/7-ansi-editors/36-pablodraw-setup

16 18 19 Moebius ANSI Art Editor
https://blocktronics.github.io/moebius/

17 23 GitHub - blocktronics/moebius: Modern ANSI & ASCII Art Editor
https://github.com/blocktronics/moebius

20 21 22 24 25 26 27 123 GlyphDrawing.Club -blog
https://blog.glyphdrawing.club/moebiusxbin-ascii-and-text-mode-art-editor-with-custom-font-support/

28 README.md - icy_draw - [fork] icy_draw is the successor to mystic ...
https://git.drkhsh.at/icy_draw/file/crates/icy_term/README.md.html

29 README.md - icy_draw - [fork] icy_draw is the successor to mystic ...
https://git.drkhsh.at/icy_draw/file/README.md.html

30 34 35 Icy Draw first public release : r/bbs
https://www.reddit.com/r/bbs/comments/176o0v7/icy_draw_first_public_release/

31 33 Let's ANSI - Icy Draw - YouTube
https://www.youtube.com/watch?v=aJ7frbjJH54

32 46 mkrueger/MysticDraw: ANSI drawing tool, supports ANSI ... - GitHub
https://github.com/mkrueger/MysticDraw

36 37 39 40 41 GitHub - cmang/durdraw: Versatile ASCII and ANSI Art text editor for drawing in the Linux/Unix/macOS terminal, with animation, 256 and 16 colors, Unicode and CP437, and customizable themes
https://github.com/cmang/durdraw

38 Durdraw - ANSI, ASCII and Unicode Art Animation Studio for Linux
https://durdraw.org/

42 Durdraw – a modern ANSI art editor for modern Unix terminals
https://news.ycombinator.com/item?id=38830687

43 Durdraw | Hackaday.io
https://hackaday.io/project/195729-durdraw

44 45 47 FreshPorts -- editors/tetradraw: Fully featured ANSI art editor/viewer with TCP/IP multidraw support
https://www.freshports.org/editors/tetradraw/

48 TundraDraw download | SourceForge.net
https://sourceforge.net/projects/tundradraw/

49 File Downloads, SAC Packs & More! - Roy/SAC
http://www.roysac.com/roy-sac_downloads_links.html

51 JavE - Java Ascii Versatile Editor
http://www.jave.de/

52 Make Your Mark - » Linux Magazine
http://www.linux-magazine.com/Issues/2009/106/JavE-ASCII-Art-Editor

53 [The Sunny Spot] ASCII-Art - Software
https://the.sunnyspot.org/asciiart/software.html

54 Monodraw for macOS — Helftone
https://monodraw.helftone.com/

55 Monodraw - ACSII art editor for drawing keyboard layouts. : r/olkb
https://www.reddit.com/r/olkb/comments/beyci9/monodraw_acsii_art_editor_for_drawing_keyboard/

56 Monodraw – a non-subscription, powerful ASCII art editor
https://news.ycombinator.com/item?id=32134469

57 59 REXPaint - A powerful and user-friendly ASCII art editor by Grid ...
https://www.gridsagegames.com/rexpaint/

58 61 REXPaint: Powerful, User-Friendly ASCII Art Editor | Hacker News
https://news.ycombinator.com/item?id=18648718

60 Visual Arts: ASCII Art: Software - Curlie
https://curlie.org/Arts/Visual_Arts/ASCII_Art/Software/

62 REXPaint hits 1.0 - Free ASCII/ANSI/Roguelike art, mapping ... - Reddit
https://www.reddit.com/r/gamedev/comments/4hmjta/rexpaint_hits_10_free_asciiansiroguelike_art/

63 REXPaint Manual
https://steveasleep.com/rexpaint_manual.html

64 ASCII Art Paint
https://kirilllive.github.io/ASCII_Art_Paint/

65 67 ASCII Art Paint
https://kirilllive.github.io/ASCII_Art_Paint/ascii_paint.html

66 ASCII Art Paint - GameDev.net
https://www.gamedev.net/projects/4396-ascii-art-paint/

68 69 70 Is there any tool to draw illustrative ascii sketch for use in code?
https://www.reddit.com/r/learnprogramming/comments/m44z3s/is_there_any_tool_to_draw_illustrative_ascii/

71 List of text editors - Wikipedia
https://en.wikipedia.org/wiki/List_of_text_editors

73 edit.tf
https://edit.tf/

74 tt-edit - teletext editor software
https://www.tt-edit.com/

75 [PDF] Your Handy Edit-TF How To Guide
https://edit.tf/doc/handy-howto.pdf

76 77 78 79 80 82 83 119 Petmate | Petmate - C64 PETSCII editor
https://nurpax.github.io/petmate/

81 An outstanding PETSCII/Screen/Sprite editing tool : r/c64 - Reddit
https://www.reddit.com/r/c64/comments/jzelay/an_outstanding_petsciiscreensprite_editing_tool/

84 87 Commodore 64 PETSCII Image - Medium
https://medium.com/@8bitsten/commodore-64-petscii-image-f608225714ec

85 86 .Markku Reunanen. » PETSCII (Marq's PETSCII Editor)
http://www.kameli.net/marq/?page_id=2717

95 MilkDrop 2 download | SourceForge.net
https://sourceforge.net/projects/milkdrop2/

96 97 MilkDrop - Wikipedia
https://en.wikipedia.org/wiki/MilkDrop

99 101 Milkdrop Preset Authoring - Paweł Porwisz
https://www.pawelporwisz.pl/winamp/Help/Windows/winamp_milkdrop_preset_authoring.php

100 Butterchurn visualizer is awesome for at home reactive visuals, JMH ...
https://www.reddit.com/r/Tipper/comments/br3evc/butterchurn_visualizer_is_awesome_for_at_home/

102 103 105 107 GitHub - projectM-visualizer/projectm: projectM - Cross-platform Music Visualization Library. Open-source and Milkdrop-compatible.
https://github.com/projectM-visualizer/projectm

104 106 projectM : r/milkdrop - Reddit
https://www.reddit.com/r/milkdrop/comments/18hw4xp/projectm/

108 jberg - NPM
https://www.npmjs.com/~jberg

109 Butterchurn Visualizer
https://butterchurnviz.com/

110 Butterchurn – A WebGL Implementation of the Milkdrop Visualizer
https://news.ycombinator.com/item?id=26813265

111 Steam Workshop::Butterchurn - Audio Visualizer
https://steamcommunity.com/sharedfiles/filedetails/?id=1857696466

115 Visualize your Music your way
https://projectm-visualizer.org/

116 117 120 124 125 126 127 Ansi drawing tools : r/bbs
https://www.reddit.com/r/bbs/comments/1687gzo/ansi_drawing_tools/

118 Text editor/viewer with ANSI codes rendering support for Windows

https://stackoverflow.com/questions/2271785/text-editor-viewer-with-ansi-codes-rendering-support-for-windows