

Graphics and Music Challenges for Classic Computer Applications

Peter Occil

This version of the document is dated 2025-10-26.

The following are three challenges I make to the computer community, relating to:

- **Graphics for classic-style game development.**
- **MIDI music synthesis for classic-style games and apps.**
- **Tileable wallpapers with limited colors and resolution.**

All three serve to arouse nostalgia among 1990s computer users.

1 Graphics Challenge for Classic-Style Games

An interesting challenge for game developers, relating to designing games with classic graphics that run on an exceptional variety of modern and recent computers.

Classic graphics here means two- or three-dimensional graphics achieved by video games from 1999 or earlier, before the *shader model* of 3-D graphics programming became mainstream among game software.¹

Most desktop and laptop computers from 2010 on, and most smartphones from 2016 on, can render even high-quality classic graphics using only software — without relying on specialized video cards — at screen resolutions typically targeted by late-1990s games.²

The specification in this challenge sets an *upper bound* on the kind of computer graphics that are of interest. Further constraints to graphics computation (such as memory, resource, color, resolution, or polygon limits) are highly encouraged.

Define the *larger screen dimension* as the larger of the screen width and the screen height.

Limit 3D graphics to the following:

1. The maximum number of polygons that can be displayed at a time is equal to screen width times screen height divided by 24.³ (Examples: $256 \times 192 / 24 = 2048$ polygons; $640 \times 480 / 24 = 12800$

¹Matt Saettler, “Graphics Design and Optimization”, Multimedia Technical Note (Microsoft), 1992, contains a rich discussion of graphics used in classic multimedia and game applications. Not mentioned in that document are graphics resembling: (1) Segmented liquid crystal displays, of the kind that Tiger Electronics was famous for. These are simple to emulate, though: design a screen-size picture that assigns each segment a unique color and, each frame, draw black pixels on the screen where the segments that are “on” are, and draw white pixels (or another background) elsewhere on the screen.(2) Vacuum fluorescent displays, notable in user interfaces of some media player applications that resemble a “stereo rack system”.

²A computer has adequate performance for classic graphics if it achieves a score of—(a) 2700 or greater on the 3DMark2000 benchmark when run without graphics acceleration, or(b) 1500 or greater on the 3DMark2000 CPU Speed Test.Either score corresponds to the running of a then-demanding 3-D graphics demo at 60 frames per second or faster.

³To bolster the suggestions in this specification, it would be of interest to find the number of polygons per frame and graphics memory use (for a given resolution and frame rate) actually achieved on average by 3-D video games in the mid- to late 1990s. Such information is hard to find and is often anecdotal. For example:(1) B. Tschirren, “Realism and Believability in MPEG-4 Facial Models”, Curtin University of Technology, 2000, includes a statement that games like *Quake III Arena* [1999] render up to 10,000 triangles per frame.(2) “A typical scene in a current [PC] application has 2000 to 2500 triangles per frame” (R. Fosner, “DirectX 6.0 Goes Ballistic With Multiple New Features And Much Faster Code”, *Microsoft Systems Journal* January

polygons.)

- A polygon is either a triangle, a convex quadrilateral, or a line segment.
 - Each vertex of the polygon must point to a vertex from the vertex list described later.
 - Each polygon can be translucent and/or wireframed.
2. The maximum number of vertices that can be used at a time is 3 times the maximum number of polygons.
 - A vertex consists of an XYZ position, an XY texture coordinate, and a red-green-blue vertex color.
 - For each color, the red component is 5 bits, the green, 5 bits, and the blue, 5 bits.
 3. Textures must have the same color format as vertex colors, or may employ a 4-, 16-, or 256-color palette with that color format. The width and height of each texture must be a power of 2. A texture's maximum width and maximum height, in pixels, are each equal to 256 or the larger screen dimension, whichever is smaller.
 4. Depth buffers (Z buffers), clear colors, and fog colors are supported.
 5. The 3D graphics buffer's resolution is the same as the screen resolution.
 6. Polygons should undergo perspective-correct texture mapping, but affine mapping may be implemented instead.⁴

Limit 2D graphics to the following: ⁵

1. Up to three 2D layers can be displayed at a time. If 3D graphics are not being displayed, a fourth 2D layer can also be displayed. Otherwise, a layer for the 3D graphics can be displayed. Each 2D layer is a rectangular array of references to *tiles* (a *tile* is a small rectangular array of pixels).
2. There are sixteen palettes of 16 colors each (using the color format for vertex colors).
3. The tiles have the same size (32×32 pixels or smaller) and each tile uses the colors of one of the sixteen palettes just described. A tile size of 8×8 pixels is suggested.
4. The 2D and 3D layers may contain transparent pixels.
5. Up to two of the 2D layers can undergo a 2D affine transformation.
6. Separate from layers, 2D sprites can be displayed. Each sprite is a rectangular array of either tiles or pixels, has size up to $X \times Y$ pixels, and may contain transparent pixels, but not translucent (semitransparent) pixels. Up to N sprites can be displayed at a time. Each sprite can be rendered above or below any of the 2D layers.
 - X and Y are each $1/4$ the larger screen dimension, rounded up to the nearest power of 2. (An alternative limit is $X = 64$ and $Y = 64$.)
 - N is calculated as $(\text{screen width} \times \text{screen height} \times 16) / (X \times Y)$, rounded up, but not more than 512.⁶
7. Tiles and sprites can be rendered flipped on either or both axes.

1999).(3) "For context, *Quake* on a Pentium Pro pumped out maybe 100K triangles/second (tris/sec.) ... at best" (M. Abrash, "Inside Xbox Graphics", *Dr. Dobbs's Journal*, August 2000); to be noted here is that the game normally ran at 320×240 pixels.(4) The **3DMark2000 benchmark** comes with two game scenes that average up to 9,400 polygons in low detail and up to 55,000 in high detail and is intended for 640×480 resolution. https://web.archive.org/web/20050404173017/http://www.nvnews.net/reviews/annihilator_pro/3dmark_2000/html/tests.htm#fill

⁴Affine (as opposed to perspective-correct) texture mapping, together with the rounding of vertex coordinates to integers and the lack of antialiasing, contributed to the characteristic distortion and instability of 3-D graphics in many PlayStation (One) games.

⁵A possible alternative to these 2D limits is to require the use of a frame buffer with no more than 8 bits per pixel and to require that all graphics be rendered in software, but I don't know of a way to describe further restrictions useful for game programming in the mid- to late 1990s style. The tile-based limits specified here also suit games that support only text display, and thus have graphics that resemble the text modes (as opposed to graphics modes) found in PCs and computer terminals.

⁶Tile- and sprite-based graphics were in place largely because they saved memory. Indeed, this system, present in the Nintendo DS and many earlier game consoles, was abandoned in the Nintendo 3DS in favor of a frame buffer.

The 3D graphics layer, if any, can be alpha blended with the 2D graphics layers in any order. ⁷

Other requirements:

- **Screen resolution:** Screen resolution is 307,200 total pixels (for example, 640×480) or smaller.⁸
- **Music:** Music is in Standard MIDI files (SMF) only.⁹ As much as possible, instruments and percussive noises should match their meanings in the General MIDI System level 1.

This specification was inspired by the graphics limitations of—

- PC games in the mid- to late 1990s,¹⁰
- home computers released before 1995,
- game consoles (handheld and for TVs) released before 2000, and
- the Game Boy Advance, Nintendo DS, and Nintendo 3DS, all of which were released after 2000 but have relatively meager graphics capability.

A game may impose further resource limits to those specifications (for example, to reduce the maximum number of 3D polygons, to disallow polygons, to reduce the number of colors per tile allowed, or **reduce to a limited set the colors**¹¹ ultimately displayed on screen). I would be interested in knowing about these limitations that a new game that adopts this document decides to impose. I would also be interested in learning about a free and open-source graphics library that implements this specification.¹² Examples of optional constraints are the following:

- The game is limited to the 16 colors of the so-called *VGA palette*.
 - In the 8-bit-per-component format, this palette’s colors are: light gray, that is, (192, 192, 192); or each color component is 0 or 255; or each color component is 0 or 128.
 - In the vertex color format, the closest colors to this palette are: 24/24/24; or each color component is 0 or 16; or each color component is 0 or 31.
- All game files can be packaged in a ZIP file or Win32 program file that takes no more than—
 - 1,457,664 bytes (the capacity of a file-allocation-table (FAT) formatted high-density 3.5-inch floppy disk), or
 - 1,213,952 bytes (the capacity of a FAT formatted high-density 5.25-inch floppy disk), or

⁷But alpha blending (the partial mixing of one color with another) was “relatively new to PC games” at the time of *Quake*’s release in 1996, according to *Michael Abrash’s Graphics Programming Black Book*. Only images with opaque and/or transparent pixels tended to be supported in early-1990s video games.

⁸If the game screen is two-level monochrome (one bit per pixel), a game could choose to allow screen resolutions up to 800,000 total pixels. For example, a 1024×768 screen has 786,432 total pixels. However, two-level monochrome graphical display modes larger than 307,200 total pixels are probably rare among consumers. The modern game *Return of the Obra Dinn* employs a two-level monochrome 800×450 display (378,000 total pixels).

⁹Standard MIDI files should be rendered using a cross-platform open-source software synthesizer (see section “Building a Public-Domain music synthesis library and instrument banks”), using either FM or wavetable synthesis; most modern PCs no longer come with hardware synthesizers. I note that it’s possible to write an FM software synthesizer supporting every MIDI instrument in less than 1024 kibibytes of code.

¹⁰This includes: (1) Windows games and screen savers written for DirectX versions earlier than 7 and using Direct3D or DirectDraw for graphics. (2) Windows games and screen savers using GDI or **WinG** for graphics and supporting Windows 98 or earlier. Examples are *Chip’s Challenge* for Windows (1992) and Brian Goble’s *The Adventures of MicroMan* (1993). (3) Games for MS-DOS or PC-9801 that were published before 2000. Examples are *Quake* (1996) and *WarCraft* (1994). (4) Games using an OpenGL version earlier than 1.2 for graphics. https://www.pcgamingwiki.com/wiki/List_of_WinG_games

¹¹<https://github.com/peteroupc/classic-wallpaper?tab=readme-ov-file#color-palettes>

¹²Especially if the library is self-contained and implements the specification with as little source code as possible. The following are examples of a graphics library that follows the spirit of this specification, even though it doesn’t necessarily meet its requirements exactly: *Tilengine, kit, DOS-like, raylib’s rlsw software renderer*. Michal Strehovský published an **interesting technique to create small game applications**. This specification does not preclude the use of prerendered graphics (as in *Space Quest 5*, *Myst*, or the original *Final Fantasy VII* on PlayStation) to simulate showing more polygons or vertices at a time than otherwise allowed. Doing this simulation using other software programming tricks would not be within the spirit of this specification, though. <https://github.com/megamarc/Tilengine> <https://github.com/rxi/kit> <https://github.com/mattiasgustavsson/dos-like> <https://github.com/raysan5/raylib> <https://migeel.sk/blog/2024/01/02/building-a-self-contained-game-in-csharp-under-2-kilobytes/>

- 730,112 bytes (the capacity of a FAT formatted normal-density 3.5-inch floppy disk), or
- 362,496 bytes (the capacity of a FAT formatted “360K” 5.25-inch floppy disk), or
- 681 million bytes (slightly less than the maximum capacity of a formatted CD-ROM).
- The game uses no more than 16 million bytes of system memory at a time.
- The game uses no more than 655,360 bytes of system memory (plus 262,144 bytes of additional memory for graphics use only) at a time.¹³
- The game aims for a frame rate of 30 frames per second.
- The game’s graphics are rendered in software. That is, the rendering does not rely on a video card, a graphics accelerator chip, or the operating system’s graphics programming interface (such as GDI, OpenGL, or Direct3D) with the sole exception of sending a finished frame buffer to the screen (such as through GDI’s `StretchDIBits` or copying to VGA’s frame buffer).
- The game’s graphics rendering employs only 32-bit and smaller integers and fixed-point arithmetic.¹⁴

1.1 Notes on Specifications

Screen resolution:

- Screen resolutions larger than 307,200 total pixels (such as 800×600) are not within the spirit of this challenge, even though more demanding games in the late 1990s, as well as the *PC 98 System Design Guide* (1997), aimed for such resolutions for 3-D graphics.
- Screen resolutions that have been used in classic games include:¹⁵
 - Video graphics array (VGA) display modes: 640×480 ,¹⁶ 320×240 ,¹⁷ 320×200 .¹⁸
 - 4:3 aspect ratio: 640×480 ,¹⁹ 512×384 ,²⁰ 400×300 ,²¹ 320×240 ,²² 256×192 ,²³ 160×120 .²⁴
 - Game console aspect ratios: 640×448 ,²⁵ 320×224 ,²⁶ 256×224 ,²⁷ 256×240 ,²⁸ 240×160 ,²⁹ 160×144 .³⁰
 - PAL home computer aspect ratios: 320×256 ,³¹ 640×512 ,³² 360×288 .³³

¹³MS-DOS applications are normally limited to 640 kibibytes or less of *conventional memory*, along with whatever memory is carried by the video card. 262,144 bytes is the usual minimum of graphics memory for VGA video cards.

¹⁴It wasn’t until the Pentium processor’s advent that floating-point arithmetic was embraced in 3-D game programming; for example, see chapter 63 of *Michael Abrash’s Graphics Programming Black Book*.

¹⁵In addition to the resolutions shown here, there are modern games that employ low resolutions with a 16:9 aspect ratio, to take advantage of high-definition displays. These include 640×360 (*Blasphemous*); 400×225 (*Unsung*); 480×270 (*Enter the Gungeon*); 320×180 (*Celeste*).

¹⁶VGA mode 12h (16 colors).

¹⁷PlayStation (One); Nintendo 3DS lower screen; larger VGA “mode X” (256 colors).

¹⁸Commodore 64; NEC PC-8001; VGA mode 13h (256 colors); Color/Graphics Adapter (CGA) 4-color mode; Atari ST 16-color mode.

¹⁹VGA mode 12h (16 colors).

²⁰One commonly supported “super-VGA” mode, especially in mid-1990s gaming, and which was also required by the *PC 98 System Design Guide*.

²¹One low resolution required by the *PC 98 System Design Guide*.

²²PlayStation (One); Nintendo 3DS lower screen; larger VGA “mode X” (256 colors).

²³Nintendo DS; NEC PC-6001; Sega Master System/Sega Mark III; MSX; Colecovision.

²⁴Rarely used VGA display mode.

²⁵PlayStation 2 NTSC.

²⁶PC Engine/TurboGrafx 16; Sega Mega Drive/Sega Genesis; Neo Geo NTSC.

²⁷Effective resolution of Famicom/Nintendo Entertainment System NTSC; Super Famicom/Super Nintendo Entertainment System NTSC.

²⁸Nintendo Entertainment System PAL; Super Nintendo Entertainment System PAL.

²⁹Game Boy Advance.

³⁰Game Boy, Game Boy Color, Sega Game Gear.

³¹Amiga PAL; Neo Geo PAL.

³²PlayStation 2 PAL. Note that this covers more than 307,200 total pixels.

³³PAL overscan.

- Two-level monochrome graphics: 720×348 ,³⁴ 640×200 ,³⁵ 512×342 .³⁶
- Enhanced Graphics Adapter aspect ratio: 640×350 .³⁷
- 8:5 aspect ratio: 640×400 ,³⁸ 320×200 .³⁹
- Other: 280×192 ,⁴⁰ 480×272 ,⁴¹ 512×424 ,⁴² 400×240 ,⁴³ 384×224 ,⁴⁴ 160×200 ,⁴⁵ 480×240 .⁴⁶

This is not a complete list. Arcade machines of the 1990s tended to vary greatly in their screen resolutions, and some game consoles, such as the Sega Saturn or Nintendo 64, allowed games to be shown in multiple screen resolutions.

Music and sound:

- Besides the limitation on music, this specification has no further limitations on sounds. Early game consoles supported sound only through one or more *programmable sound generators*, such as square and triangle wave generators, as opposed to digitized sound (pulse-code modulation, such as .WAV files). Games that choose to constrain file size may wish to implement software versions of programmable sound generators for at least some of their sounds. When digitized sounds are supported in classic games, they typically have a sample rate of 8000, 11,025, 22,050, or 44,100 Hz, are either mono or stereo, and take 8 or 16 bits per sample.⁴⁷

Frame rate:

- Classic games for desktop or laptop computers tended to aim for a frame rate of 30, 40, or 60 frames per second, and game consoles for TVs were designed for the TV’s usual refresh rate (nearly 60 frames per second for NTSC and 50 for PAL).⁴⁸

Memory:

- This specification does not impose a limit on graphics memory use (akin to the video memory, or VRAM, of a video card). One suggested example, given in kibibytes of graphics memory, is the screen width times screen height divided by 24, which is slightly less than 13.2 million bytes for 640×480 resolution. (A kibibyte is 1024 bytes.) Imposing a limit on graphics memory use does not limit the size or number of textures, 3-D models, or other graphics files a game can have.⁴⁹

³⁴Hercules Graphics Card two-level monochrome.

³⁵Color/Graphics Adapter (CGA) two-level monochrome; NEC PC-8801 8-color mode; Atari ST 4-color mode.

³⁶12-inch classic Macintosh.

³⁷16 colors.

³⁸NEC PC-9801 8-color mode; Atari ST two-level monochrome.

³⁹Commodore 64; NEC PC-8001; VGA mode 13h (256 colors); Color/Graphics Adapter (CGA) 4-color mode; Atari ST 16-color mode.

⁴⁰Apple II.

⁴¹PlayStation Portable.

⁴²MSX 2.

⁴³Effective resolution of Nintendo 3DS upper screen without parallax effect.

⁴⁴Virtual Boy.

⁴⁵CGA 16-color mode.

⁴⁶Minimum resolution for “handheld PCs” (*Windows CE Programmer’s Guide*, MSDN Library, June 1998).

⁴⁷The *Multimedia PC Specification* (1991) required support in “multimedia PCs” for rendering at least 8-bit-per-sample mono digitized sound at 11,025 and 22,050 Hz. The Multimedia PC level 2 specification (1993) required support in “multimedia PCs” for rendering at least 16-bit-per-sample stereo digitized sound at 44,100 Hz.

⁴⁸Until the early 1990s, the number of pixels rendered per second was usually small, limiting the supported size and frame rate for arbitrary video content. Indeed, for example, the *Multimedia PC Specification* (1991) recommended that video cards be able to transfer up to 8-bit-per-pixel graphics at a rate of 140,000 pixels per second or faster given 40 percent of CPU bandwidth. The Multimedia PC level 2 specification (1993) upped this recommendation to 1.2 million pixels per second (sufficient for 320×240 pixel video at 15 frames per second, the recommendation in article Q139826, “AVI Video Authoring Tips & Compression Options Dialog Box”, 1995). For details on these specifications, see article Q106055 in the Microsoft Knowledge Base. Both recommendations are far from the 6.144 million pixels per second needed to display 640×480 pixel video smoothly at 20 frames per second.

⁴⁹PC games released in 1999 tended to require 32 million bytes of system memory. Meanwhile, *Quake* (1996) required 8 million and recommended 16 million bytes of system memory. It is worth noting that, before 1995, computer memory was expensive,

2 Building a Public-Domain music synthesis library and instrument banks

To improve support for MIDI (Musical Instrument Digital Interface) music playback in open-source and other applications, I challenge the community to write the following items, all of which must be released to the public domain or under the Unlicense.

- A cross-platform open-source library for *software* synthesis (translation into digitized sound such as PCM) of MIDI data stored in standard MIDI files (SMF, .mid), using instrument sound banks (synthesizer banks) in SoundFont 2 (.sf2), Downloadable Sounds (.dls), and in OPL2, OPL3, and other FM synthesis sound banks, and possibly also in Timidity++/UltraSound patch format (.cfg, .pat). (Similar to *Fluidsynth*, but in the public domain or under the Unlicense. Instrument sound banks are files that describe how to render MIDI instruments as sound. In addition, the source code in the nonpublic-domain *foo_midi*, *libADLMIDI*, *libOPNMIDI*, *OPL3BankEditor*, and *SpessaSynth* may be useful here, but review their licenses first.)
 - The library should support popular loop-point conventions found in MIDI files.
 - The library should support seeking of MIDI files such that a pause and resume function can be offered by a media player.
- An instrument sound bank for wavetable synthesis of all instruments and percussive noises in the General MIDI System level 1 specification.
 - Instruments should correspond as closely as possible to those in that specification, but should be small in file size or be algorithmically generated.
 - Instruments can be generated using the public-domain single-cycle wave forms found in the AdventureKid Wave Form collection, found at: **AKWF-FREE**⁵⁰.
 - The samples for each instrument may, but need not be, generated by an algorithm, such as one that renders the instrument’s tone in the frequency domain. An example of this is found in **com.sun.media.sound.EmergencySoundbank**⁵¹, which however is licensed under the GNU General Public License version 2 rather than public domain.
 - The instrument sound bank should be in either SoundFont 2 (.sf2) or Downloadable Sounds (.dls) format.⁵²
 - The volume of all instruments in the sound bank should be normalized; some instruments should not sound louder than others.
- An instrument sound bank for FM synthesis of all instruments and percussive noises in the General MIDI System level 1 specification. Instruments should correspond as closely as possible to those in that specification.

3 Other Challenges

- **Classic desktop wallpaper:** See the “**peteroupc/classic-wallpaper**”⁵³ repository for a challenge on creating tileable desktop wallpapers with a limited palette of colors and a limited pixel size — such wallpapers are getting ever harder to find because desktop backgrounds today tend to cover the full computer screen, to employ thousands of colors, and to have a high-definition resolution (1920 × 1080 or larger).
- **Button and border styles:** See **uielements.md** in the “**peteroupc/classic-wallpaper**” repository

so that computers with more than 4096 kibibytes of system memory (and 1024 kibibytes of video memory) were rare among consumers.

⁵⁰<https://github.com/KristofferKarlAxelEkstrand/AKWF-FREE>

⁵¹<https://github.com/apple/opensource/blob/xcodejdk14-release/src/java.desktop/share/classes/com/sun/media/sound/EmergencySoundbank.java>

⁵²A sound bank of decent quality in either format is about 4 million bytes in size. Making these banks would be easier if there were a guide on producing decent-quality instrument banks from the recordings of real musical instruments (rather than copying or converting other instrument banks or recording from commercial synthesizers).

⁵³<https://github.com/peteroupc/classic-wallpaper>

tory⁵⁴ for a challenge on writing computer code (released to the public domain or under the Unlicense) to draw button and border styles for classic graphical user interfaces.

4 License

Any copyright to this page is released to the Public Domain. In case this is not possible, this page is also licensed under **Creative Commons Zero**⁵⁵.

5 Notes

⁵⁴<https://github.com/peteroupc/classic-wallpaper/blob/main/uielements.md>

⁵⁵<https://creativecommons.org/publicdomain/zero/1.0/>