Graphics and Music Challenges for Classic-Style Computer Applications

Peter Occil

This version of the document is dated 2026-02-16.

The following are 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**[1].
- **Other challenges and projects**.

All may interest 1990s computer users.

1 Contents

2 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.

[1] https://github.com/peteroupc/classic-wallpaper

In this document, *classic graphics* generally means two- or three-dimensional graphics achieved by video games from 1999 or earlier, before the advent of programmable "shaders". For details, see "**Classic Graphics in Scope**", later. (To summarize: In general, a game screen of 640 × 480 or smaller, up to 12,800 3-D polygons at a time [fewer if the game screen is smaller], and tile- or sprite-based 2-D graphics are involved.)

This challenge is intended to encourage the making of video games with very low resource requirements (say, no more than 64 million bytes of system memory); most desktop and laptop computers from 2010 on, and most smartphones from 2016 on, can draw even high-quality pre-2000 graphics using only software — without relying on specialized video cards — at 640 × 480 pixels or smaller, screen resolutions typically targeted by video games in the 1990s and earlier.[2]

The 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 triangle limits) are highly encouraged. It is also encouraged to write a free and open-source graphics engine[3] or establish a lean programming interface for this graphics specification. For details, see "**Lean Programming Interfaces for Classic Graphics**[4]".

2.1 The Specification

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

Limit 3-D graphics to the following:[5]

1. The maximum number of primitives that can be displayed at a time (per frame) is equal to screen width times screen height divided by 24. (See also survey project in "Other Challenges and Projects", later.)
    - A *primitive* is either a triangle or a line segment. An application may also consider a convex quadrilateral to be a primitive.
    - Each vertex of the primitive points to a vertex from the vertex list described later.
    - Each primitive can be translucent.
2. The maximum number of vertices that can be displayed at a time is 3 times the maximum number of primitives.
    - A *vertex* consists of an XYZ position, an XY texture coordinate, and a red–green–blue vertex color.
    - Each vertex color follows this color format: The red, green, and blue components occupy up to 5 bits each.[6]
3. Each *texture* (an image that is applied to the surface of 3-D objects)—
    - is in a 16-bit-per-pixel format, where each pixel has the vertex color format given earlier, or

[2]A computer has adequate performance for classic graphics if it achieves a score of—(a) 3108 or more 3D marks on the 3DMark2000 benchmark (640 × 480) when run without graphics acceleration, or(b) 195 or greater on the 3DMark2000 CPU speed test.Both figures correspond to the running of two graphically demanding 3-D demos, at three levels of detail each, at 60 frames per second (adjusted downward as needed if a demo's detail level averages more than 12,800 triangles per frame; see the section "Test Descriptions" in the 3DMark2000 help).The games this challenge encourages are those with very low resource requirements, namely those that can run with acceptable performance even on very low-end computers, say, those that date from 2010 or earlier or support Windows 7, Windows XP, or an even older operating system. In this sense, even though today's Unity and Unreal game engines can allow the making of games with "classic graphics", they are far from lightweight and their use is not within the spirit of this challenge.

[3]The following are examples of a graphics library that follows the spirit, even if not the letter, of this specification: ***Tilengine***, ***kit***, ***DOS-like***, ***raylib*'s `rlsw` software renderer**. Michal Strehovský published an **interesting technique to create small game applications**, and so did **Jani Peltonen**.I offer a **template for Win32 applications** that supports 8- and 24-bit-per-pixel game images and paints finished game images using `StretchDIBits`. 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/ https://www.codeslow.com/2019/12/tiny-windows-executable-in-rust.html https://gist.github.com/peteroupc/f1a5d8e45e27123b86b284271cfd802b

[4]https://peteroupc.github.io/graphicsapi.html

[5]One editor specialized for creating classic 3-D models is the open-source tool ***Blockbench***. https://www.blockbench.net/

[6]If there is interest, this format may instead be: The red and blue components occupy 5 bits each; the green component, 6 bits.

- is in a 1-, 2-, 4-, or 8-bit-per-pixel format and has a table of colors with the vertex color format given earlier.

4. The width and height of each texture is a power of 2.
5. A texture's maximum width and maximum height, in pixels, are each equal to 256 or the larger screen dimension, whichever is smaller.
6. Textures may contain transparent pixels.
7. Depth buffers (Z buffers) and depth-based pixel-level fog are supported.
8. The 3-D graphics buffer's resolution is the same as the screen resolution.
9. 3-D primitives should undergo perspective correction, but this is optional.[7]

   Example: For a "screen resolution" (see later) of 640 × 480 pixels, no more than 12,800 primitives (640 × 480 / 24) and 38,400 vertices can be shown at a time, and the maximum texture size is 256 × 256 pixels.

Limit 2-D graphics to the following: [8]

1. Up to three 2-D *layers* can be displayed at a time. If 3-D graphics are not being displayed, a fourth 2-D layer can also be displayed. Otherwise, a layer for the 3-D graphics can be displayed. Each 2-D layer is a rectangular array of references to *tiles* (a *tile* is a small rectangular array of pixels).
2. Up to two of the 2-D layers can undergo a 2-D affine transformation.
3. The tiles have the same size (32 × 32 pixels or smaller). A tile size of 8 × 8 pixels is suggested.
4. The application chooses one:
   1. Each tile is in a 1-bit-per-pixel format and uses one of 16 *color tables*, with 2 colors per table.
   2. Each tile is in a 2-bit-per-pixel format and uses one of 16 color tables, with 4 colors per table.
   3. Each tile is in a 4-bit-per-pixel format and uses one of 16 color tables, with 16 colors per table.
   4. There is a single 256-color table for use by tiles. Each tile is in an 8-bit-per-pixel format.
5. Each color in each color table used by tiles is of the vertex color format given earlier.
6. Tiles can be horizontally flipped, vertically flipped, or both.
7. Separate from layers, 2-D *sprites* can be displayed. Each sprite is a rectangular array of either tiles or pixels.
8. Each sprite has size up to X × Y pixels, where 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.)
9. Up to N sprites can be displayed at a time, where N is calculated as (screen width × screen height × 16) / (X × Y), rounded up, but not more than 512.[9]
10. Each sprite made of pixels (rather than tiles) has a pixel format allowed for 3-D textures, given earlier.
11. Each sprite can be drawn above or below any of the 2-D layers.
12. The application chooses one:
    1. Each sprite can undergo a 2-D affine transformation.
    2. Each sprite can be horizontally flipped, vertically flipped, or both.[10]
    3. No affine transformation or flipping of sprites is allowed.

[7] Perspective correction accounts for distance from the viewer: closer objects appear larger. The lack of perspective correction (as in what is called *affine texture mapping*), together with the rounding of vertex coordinates to integers and the lack of smoothing (antialiasing) of edges, contributed to the characteristic distortion and instability of 3-D graphics in many PlayStation (One) games.

[8] A possible alternative to these 2-D limits is to require the use of a frame buffer (array of color samples, called pixels, in computer memory) with no more than 8 bits per pixel (no more than 256 simultaneous colors) and to require that all graphics be *rendered in software* (see section "Optional Limits"), 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.

[9] Tile- and sprite-based graphics were in place largely because they saved memory; they were popularized by the arcade game *Galaxian*. 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.

[10] SEGA arcade machines from the 1980s and earlier had rudimentary systems for scaling (stretching or shrinking) sprites horizontally and vertically. In the Super Famicom/Super Nintendo Entertainment System, sprites could not be scaled, but they could be flipped.

13. Layers, tiles, and sprites may contain transparent pixels, but not translucent (semitransparent) pixels. As an exception, the 3-D layer may contain translucent pixels.[11]

The 3-D graphics layer, if any, can be alpha blended with the 2-D graphics layers in any order. [12]

> **Example:** For a "screen resolution" (see later) of 640 × 480 pixels, one choice is: 4-bit-per-pixel tiles, 8 × 8 tiles, sprites up to 160 × 160 pixels, no more than 192 sprites at a time, and no flipping or transformation of sprites.

Other requirements:

- **Screen resolution:** The game screen image has no more than 307,200 total pixels (for example, 640 × 480, or 640 pixels horizontally and 480 pixels vertically).[13]
- **Music:** Music is in Standard MIDI files (SMF) only. The General MIDI System level 1 should be followed for such files.[14]

2.2 Classic Graphics in Scope

This specification for "classic graphics"[15] in modern games largely reflects the graphics limitations of—

- consumer PCs (personal computers) released in the mid- to late 1990s,
- home computers released before 1995,
- game consoles (handheld and for TVs) released before 2000,
- arcade machines with similar performance to machines described earlier, and
- the Game Boy Advance, Nintendo DS, and Nintendo 3DS, all of which were released after 2000 but have relatively meager graphics ability.

In addition, video-game graphics for personal digital assistants, graphical calculators, and cellular phones (generally those released before 2007) are within the spirit of this specification, up to the performance of consumer PCs released before 2000.

[11] If there is interest I may allow 2-D sprites to have translucent pixels in this specification. But support for such sprites was probably rare before 1995.

[12] But alpha blending (the mixing of one image 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.

[13] If the game screen image uses two colors only (such as black and white), the game could choose to allow it to have up to 800,000 total pixels. For example, a 1024 × 768 display has 786,432 total pixels. However, two-color 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-color 800 × 450 display (378,000 total pixels).In the Godot engine, the screen resolution corresponds to the "Viewport Width" (`window/size/viewport_width`) and "Viewport Height" (`window/size/viewport_height`) project settings. In the Unity engine, the screen resolution corresponds to the settings `defaultScreenWidth` and `defaultScreenHeight`; in Unreal Engine, apparently the settings `ResolutionSizeX` and `ResolutionSizeY`. But a lighter-weight graphics engine than Unity, Unreal, or even Godot would better suit the spirit of this specification.

[14] Standard MIDI files should be played back using a cross-platform open-source software synthesizer (see section "Building a Public-Domain music synthesis library and instrument banks"), using either FM or wave-table 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.Standard MIDI files organize MIDI sounds into up to 16 *channels*, each occupied by at most one "instrument" at a time. Under the *Multimedia PC Specification* (1992), the first ten channels were intended for high-end synthesizers (where the tenth is percussion); the thirteenth through sixteenth, for low-end ones (sixteenth is percussion), and the nonpercussion channels were arranged in decreasing order of importance. This convention was abandoned with the rise in support for the General MIDI System level 1 (see Q141087, "DOCERR: MarkMIDI Utility Not Provided in Win32 SDK", in the Microsoft Knowledge Base): now all 16 channels are supported (with only the tenth for percussion) and need not be arranged by importance.

[15] Matt Saettler, "Graphics Design and Optimization", Multimedia Technical Note (Microsoft), 1992, contains a rich discussion of graphics used in computer games and other audiovisual computer applications up to 1992. 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 image that assigns each segment a unique color and, each frame, draw black where where the segments that are "on" are, and draw white (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".

Some video game hardware from the late 1990s may have 3-D graphics capabilities beyond what is "classic" here, such as SEGA Model 3 (1996), SEGA NAOMI (1998), or NVIDIA GeForce 256 (late 1999).

In general, PC applications that feature classic graphics include:

1. Windows applications written for DirectX versions earlier than 7 and using Direct3D or DirectDraw for graphics.
2. Windows games using GDI or **WinG**[16] 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), *WarCraft* (1994), and the first titles of the Touhou Project series (1997-1998).
4. Games using an OpenGL version earlier than 1.2 for graphics.
5. So-called "multimedia titles" from the 1990s, or applications resembling interactive versions of books (generally reference and other nonfiction works), complete with sound, animation, and video. See the *Authoring Guide* that came with Microsoft's Multimedia Development Kit.

One of the following games can be considered an upper limit to what is considered "classic graphics" in this specification.

- *Quake III Arena* (December 1999), which **required DirectX 7 and at least 64 million bytes of memory**[17].
- *Falcon 4.0* (1998).

2.3 Optional Limits

A game may impose further constraints to this specification (for example, to reduce the maximum number of 3-D triangles, to disallow 3-D graphics, to reduce the number of colors per tile allowed, or to **reduce to a limited set the colors**[18] ultimately displayed on screen). I would be interested in knowing about these limitations that a new game that adopts this document decides to impose.

Examples of optional constraints are the following:

- The game displays no more than 16 colors at a time.[19]
- The game is limited to the 16 colors of the so-called *VGA palette*.
  - In the 8-bit-per-component color 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
  - 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
  - 65,536 bytes,[20] 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.[21]

[16] https://www.pcgamingwiki.com/wiki/List_of_WinG_games

[17] https://www.dosdays.co.uk/topics/early_3d_games.php

[18] https://github.com/peteroupc/classic-wallpaper?tab=readme-ov-file#color-palettes

[19] An example is *Loom* (1990).

[20] Popular file size limit of so-called "64K intros".

[21] MS-DOS applications are normally limited to 640 kibibytes or less of *conventional memory*, along with whatever memory is carried by the video card. (PCs running on the very early Intel 8086 and 8088 processors can map out no more than 640

- The game is a Win32 application compatible with Windows XP.
- The game is a Win32 application compatible with Windows 98.
- The game aims for a rate of 30 frames per second.
- The game's graphics are *rendered in software*. This means that the rendering of graphics does not rely on a video card, a graphics accelerator chip, or the operating system's graphics API (such as GDI, OpenGL, or Direct3D) with the sole exception of sending a finished game screen image to the player's display (such as through GDI's `StretchDIBits` or copying to VGA's video memory).
- The game's graphics rendering employs only 32-bit and smaller integers and fixed-point arithmetic.[22]
- The game renders only one-unit-thick white line segments on a black background (or vice versa), and displays no more than 320 of those segments at a time.

2.4 Notes on Specification

This section has notes on this specification, such as how its requirements correspond to the graphics abilities of classic video games.

2.4.1 Screen resolutions

- 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:[23]
  - Video graphics array (VGA) display modes: 640 × 480,[24] 320 × 240,[25] 320 × 200.[26]
  - 4:3 aspect ratio: 640 × 480,[27] 512 × 384,[28] 400 × 300,[29] 320 × 240,[30] 256 × 192,[31] 160 × 120.[32]
  - Game console aspect ratios: 640 × 448,[33] 320 × 224,[34] 256 × 224,[35] 256 × 240,[36] 240 × 160,[37] 160 × 144.[38]
  - 5:4 aspect ratio:[39] 320 × 256,[40] 360 × 288.[41]

kibibytes of system memory.) 262,144 bytes is the usual minimum of graphics memory for VGA video cards.

[22]It 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*.

[23]In addition to the resolutions shown here, there are modern games that employ low resolutions with the same 16:9 aspect ratio as high-definition displays. These include 640 × 360 (*Blasphemous*); 400 × 225 (*Unsighted*); 480 × 270 (*Enter the Gungeon*); 320 × 180 (*Celeste*).

[24]VGA mode 12h (16 colors).

[25]PlayStation (One); Nintendo 3DS lower screen; larger VGA "mode X" (256 colors).

[26]Commodore 64; NEC PC-8001; VGA mode 13h (256 colors); Color/Graphics Adapter (CGA) 4-color mode; Atari ST 16-color mode; **Amiga NTSC**. https://blog.johnnovak.net/2022/04/15/achieving-period-correct-graphics-in-personal-computer-emulators-part-1-the-amiga

[27]VGA mode 12h (16 colors).

[28]One commonly supported "super-VGA" mode, especially in mid-1990s gaming, and which was also recommended by the *PC 98 System Design Guide*.

[29]One low resolution recommended by the *PC 98 System Design Guide*.

[30]PlayStation (One); Nintendo 3DS lower screen; larger VGA "mode X" (256 colors).

[31]Nintendo DS; NEC PC-6001; SEGA Master System/SEGA Mark III; MSX; Colecovision.

[32]Rarely used VGA display mode.

[33]PlayStation 2 NTSC.

[34]SEGA Mega Drive/SEGA Genesis; Neo Geo NTSC.

[35]Effective resolution of Famicom/Nintendo Entertainment System NTSC; Super Famicom/Super Nintendo Entertainment System NTSC; minimum resolution of PC Engine/TurboGrafx 16.

[36]Nintendo Entertainment System PAL; Super Nintendo Entertainment System PAL.

[37]Game Boy Advance.

[38]Game Boy, Game Boy Color, SEGA Game Gear.

[39]Aspect ratio found above all in PAL (phase-alternating-line) displays. The resolution 640 × 512 (PlayStation 2 PAL), included in this category, covers more than 307,200 total pixels.

[40]Amiga PAL (same pixel spacing horizontally as vertically); Neo Geo PAL.

[41]PAL overscan.

  - Two-color graphics: 720 × 348,[42] 640 × 200,[43] 512 × 342.[44]
  - Enhanced Graphics Adapter aspect ratio: 640 × 350.[45]
  - 8:5 aspect ratio: 640 × 400,[46] 320 × 200.[47]
  - Other: 280 × 192,[48] 480 × 272,[49] 512 × 424, [50] 400 × 240,[51] 384 × 224,[52] 160 × 200,[53] 480 × 240.[54]

  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 alter the screen resolution during gameplay.

- As of early 1997, "[s]urveys indicate[d] that the great majority of [PC] users operate[d] in 640[×]480 resolution with 256 colors".[55]

- A game can support—

  - multiple sizes for the area of the screen where the game's action is drawn, or
  - pixel-column or -row doubling,

  or both features, without changing the size of the game's image. For example, the original *Doom* (1993) supported several sizes of this kind (on PC, they were 96 × 48, 128 × 64, 160 × 80, and so on up to 288 × 144, as well as 320 × 168 and 320 × 200) and optional pixel-column doubling.[56]

- Games within the scope of this challenge are meant to be run in a desktop window if the player's display is 800 × 600 pixels or larger. The same is true if the game's resolution is 620 × 420 or smaller and the player's display is 640 × 480. The game may also support full-screen display.

2.4.2 Frame rate

- No particular frame rate is required.[57]

- Modern games implementing this specification can choose to target a frame rate typical of today, such as 30, 40, or 60 frames per second.

- Game consoles for TVs were designed for how often TVs can draw their image (nearly 60 frames per second for NTSC and 50 for PAL).

[42] Hercules Graphics Card two-color.
[43] Color/Graphics Adapter (CGA) two-color; NEC PC-8801 8-color mode; Atari ST 4-color mode.
[44] 12-inch classic Macintosh.
[45] 16 colors.
[46] NEC PC-9801 8-color mode; Atari ST two-color.
[47] Commodore 64; NEC PC-8001; VGA mode 13h (256 colors); Color/Graphics Adapter (CGA) 4-color mode; Atari ST 16-color mode; **Amiga NTSC**. https://blog.johnnovak.net/2022/04/15/achieving-period-correct-graphics-in-personal-computer-emulators-part-1-the-amiga
[48] Apple II.
[49] PlayStation Portable.
[50] MSX 2.
[51] Effective resolution of Nintendo 3DS upper screen without parallax effect.
[52] Virtual Boy.
[53] One "Tandy graphics adapter" mode.
[54] Minimum resolution for "handheld PCs" (*Windows CE Programmer's Guide*, MSDN Library, June 1998).
[55] S. Pruitt, "Frequently Asked Questions About HTML Coding for Internet Explorer 3.0", updated Jan. 30, 1997.
[56] Fabien Sanglard, *Game Engine Black Book: Doom*.
[57] Until the early 1990s, the number of color samples (pixels) an application can transfer per second was usually small, limiting the supported size and frame rate for arbitrary video content. Indeed, for example, the *Multimedia PC Specification* (1992) recommended that video cards be able to transfer up to 8-bit-per-sample graphics at a rate of 140,000 samples per second or faster given 40 percent of CPU bandwidth. The Multimedia PC level 2 specification (1993) upped this recommendation to 1.2 million samples per second (sufficient for 320 × 240 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 samples per second needed to display 640 × 480 video smoothly at 20 frames per second.

- *Doom* (1993) operated at 35 frames per second but could not be run at that rate (under default settings) by typical PCs of the time.[58]
- For comfort reasons, a minimum frame rate may be required for video games that offer "**3-D vision**[59]" by rendering multiple views of the scene at a time, in conjunction with special glasses (for example, a SEGA Master System accessory) or a virtual-reality headset (for example, Nintendo's Virtual Boy). But such games were rare before 2000.

2.4.3 3-D graphics

- The ability to display more than 20,000 triangles at a time (per frame) is not within the spirit of this challenge, even for higher screen resolutions. Most 3-D video games before 2000 displayed well fewer than that, but there may be exceptions, such as arcade games for the SEGA Model 3.
- This specification allows for prerendered graphics (as in *Space Quest 5*, *Myst*, or the original *Final Fantasy VII* on PlayStation), to simulate showing more triangles or vertices at a time than otherwise allowed.
- This specification allows for drawing a 3-D graphic as a ***voxel mesh***[60] (formed from point samples in 3-D, rather than 2-D, called *voxels*), as long as the triangle limits are respected. Ways to render voxel meshes without relying on triangles (such as by layers of sprites) are outside the spirit of this specification unless the meshes are *rendered in software* (see section "Optional Limits").
- It wasn't until 1995 that 3-D video cards became widely available for consumer PCs.[61] In 3-D video games for PCs "[i]n 1995/1996, it was not uncommon to have 30-50% of the game screen filled with polygons without textures" (according to an **article**[62] that compared *Havoc* [1995] with *Mortal Kombat 4* [1997]).
- The following 3-D graphics capabilities, typical of the late 1990s, are within the spirit of this specification: Z buffering (depth buffering), bilinear filtering, flat shading, Gouraud shading, perspective correction,[63] per-vertex specular highlighting, per-vertex depth-based fog, line drawing, two-texture blending, edge antialiasing (smoothing), MIP mapping, source alpha blending, and destination alpha blending.[64] Software that is as performant as hardware meeting the requirements and recommendations of the *PC 99 System Design Guide* sections 14.27 to 14.34, except for the screen resolution, frame rate, and double buffering requirements, is recommended. Stencil buffers, bump mapping, environment mapping, and three- or four-texture blending are borderline "classic-graphics" capabilities. Bilinear filtering and edge antialiasing are optional under this specification.
- Phong shading (pixel-level specular highlighting) is not within the spirit of this specification, given that it was too slow for real-time graphics as of 2000's beginning.
- This specification is not centered on video games that offer "3-D vision" (see note under "Frame rate"), given how rare they were before 2000.

---

[58] Fabien Sanglard, *Game Engine Black Book: Doom*.

[59] https://www.pcgamingwiki.com/wiki/Glossary:Native_3D

[60] https://blog.danielschroeder.me/blog/voxel-renderer-objects-and-animation

[61] By contrast, 3-D video cards have been offered for professional-use computers since the mid-1980s; the first such cards for PCs that supported real-time display were **introduced in 1988**. https://retro.swarm.cz/sgi-irisvision-add-in-3d-accelerator-for-pc-1990/

[62] https://retro.swarm.cz/s3-virge-325-vx-dx-gx-gx2-series-of-early-3d-accelerators-deep-dive/

[63] Perspective correction accounts for distance from the viewer: closer objects appear larger. The lack of perspective correction (as in what is called *affine texture mapping*), together with the rounding of vertex coordinates to integers and the lack of smoothing (antialiasing) of edges, contributed to the characteristic distortion and instability of 3-D graphics in many PlayStation (One) games.

[64] *Quake* (1996) also employed *subdivision rasterization* for drawing small and relatively distant triangles whose vertices are rounded to integers, an algorithm likewise in scope here (*Michael Abrash's Graphics Programming Black Book*, chapter 69).

2.4.4 Screen image effects (filters)

- Effects that modify the game screen image to emulate CRT displays[65] are outside the scope of this challenge. So are effects that **scale**[66] the game screen to fit the height or width of the player's display.[67] This specification assumes those effects are not in place. A game can have those effects if it wishes, but they should be in-game settings.

2.4.5 Sounds

- 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 sounds[68]. 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.[69]

2.4.6 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.[70]
- Before 1995, computer memory was expensive, so that computers with more than 4096 kibibytes of system memory (and 1024 kibibytes of video memory) were rare among consumers; see "**Typical PCs Each Year**[71]".
- Before 1999, computers with more than 32,768 kibibytes of system memory were rare among consumers. (In the *PC 99 System Design Guide*, "entertainment PCs" required at least 64 million bytes of system memory.)

3 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.

[65] CRT displays, or cathode-ray-tube displays, were the typical kind of computer monitors and TVs in the 1980s and 1990s.

[66] https://www.pcgamingwiki.com/wiki/Glossary:Scaling

[67] Effects to scale the game screen include so-called "pixel-art scaling algorithms" such as `HQX` and `2xSaI`, as well as bilinear or nearest-neighbor filtering.Effects to scale the game screen do not include the decoding of small videos to fit the *game screen*, as opposed to the player's display. It was common for 1990s games to have videos smaller than the game screen and to scale those videos to fit the game screen "on the fly", in the process of displaying them. For example, such a game could decode videos of size 160x100 to fit a game screen of 320 × 200. (See, for instance, Nigel Thompson, "**Stretching 256-Color Images Using Interpolation**)", Microsoft Developer Network, March 7, 1995.) https://learn.microsoft.com/en-us/previous-versions/ms969922(v=msdn.10

[68] Digitized sound is also known as pulse-code modulation (PCM) and is often stored in files ending in ".WAV".

[69] The *Multimedia PC Specification* (1992) required support in "multimedia PCs" for playback of 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 playing back at least 16-bit-per-sample stereo digitized sound at 44,100 Hz.

[70] PC 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.

[71] https://www.dosdays.co.uk/topics/typical_pc_per_year.php

- 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 wave-table synthesis of all instruments and percussive (drum) 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**[72].
  - 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`[73], 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. See next section on a challenge to writing a guide on sound bank development.
  - 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.

4 Other Challenges and Projects

Other challenges and projects I make to the computer community.

4.1 Classic desktop wallpaper

See the "**peteroupc/classic-wallpaper**[74]" repository for a challenge on creating tileable desktop wallpapers with a limited selection of colors and limited dimensions in pixels — 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).

4.2 Button and border styles for classic interfaces

See **"Traditional User-Interface Graphics" in the "peteroupc/classic-wallpaper" repository**[75] 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.

[72] https://github.com/KristofferKarlAxelEkstrand/AKWF-FREE
[73] https://github.com/apple/openjdk/blob/xcodejdk14-release/src/java.desktop/share/classes/com/sun/media/sound/EmergencySoundbank.jav
[74] https://github.com/peteroupc/classic-wallpaper
[75] https://github.com/peteroupc/classic-wallpaper/blob/main/uielements.md

4.3 Survey of polygon and memory usage in 1990s 3-D games

To buttress the suggestions in the **specification on classic graphics**, given earlier in this page, it would be of interest to find the number of triangles or polygons per frame and graphics memory usage (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.[76]

4.4 Sound bank development guide

Write an open-source and detailed guide on using free-of-cost software to produce decent-quality instrument banks from the recordings of real musical instruments (rather than copying or converting other instrument banks or recording from commercial synthesizers). See the section on **building instrument banks, earlier**. For this purpose, a sound bank in SoundFont 2 or Downloadable Sounds format that is of decent quality is about 4 million bytes in size.

5 License

6 Notes

[76]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 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. Dobb's Journal*, August 2000); to be noted here is that the game normally ran at a screen resolution of 320 × 240.(4) According to the help for the 3DMark2000 benchmark, that benchmark comes with two game scenes that average up to 9,400 polygons in low detail and up to 55,000 in high detail.(5) An **early study of polygon rendering rates** in DOS-based 3-D games, by B. Johanson and B. Oberstein (1996), shows the difficulty of finding triangle output rates in DOS games.

[77]https://creativecommons.org/publicdomain/zero/1.0/