Gaming Tech Glossary

Hardware basics

CPU

- what it is: the central processor that handles general game logic, physics, AI, and background tasks.
- why it matters: some games are limited by the cpu, especially big worlds and many players.
- example: in a large battle royale, your frame rate dips during explosions because the cpu is doing lots of work at once.

GPU

- what it is: the graphics processor that draws every frame.
- why it matters: most modern games are limited by the gpu at higher settings and resolutions.
- example: turning ultra shadows and ray tracing on makes the gpu the bottleneck.

APU

- what it is: a chip that combines a cpu and integrated graphics on one package.
- why it matters: good for laptops and small desktops without a separate graphics card.
- example: an entry gaming build that plays esport titles at 1080p on an APU.

NPU

- what it is: a neural processing unit designed for AI tasks.
- why it matters: used for features like background AI effects and future game tools.
- example: video apps using the NPU to run background blur while you game.

Cores and threads

- what it is: cores are independent workers inside the cpu. threads are the tasks they handle.
- why it matters: more cores help with multitasking and some game engines.

- example: streaming, voice chat, and the game run smoother on an 8 core cpu than a 4 core one.

clock speed

- what it is: how fast a cpu or gpu core runs, measured in GHz or MHz.
- why it matters: higher clocks can mean higher performance if cooling and power allow.
- example: boosting a gpu from 2.3 to 2.6 GHz can raise frame rate a little in the same game settings.

IPC

- what it is: instructions per clock. how much work a cpu core does each tick.
- why it matters: two chips at the same clock can perform differently if IPC is different.
- example: a newer cpu at 4.5 GHz can beat an older one at 5.0 GHz due to better IPC.

cache

- what it is: very fast memory inside the cpu or gpu.
- why it matters: larger or smarter cache can reduce stutter and improve average frame rate.
- example: a cpu with extra cache keeps game data close and reduces slow downs in a big open world.

VRAM

- what it is: video memory on the graphics card.
- why it matters: high resolution textures and ray tracing need more VRAM.
- example: using 4K textures on a card with low VRAM can cause hitching as data swaps in and out.

Memory bandwidth and bus width

- what it is: how fast data moves from VRAM to the gpu, and how wide the path is.
- why it matters: affects high resolution performance and heavy texture scenes.
- example: a card with wider bus width and faster memory feeds the gpu better at 4K.

Power, TDP, and TGP

- what it is: power draw guidelines for cpu and gpu.
- why it matters: higher power allows higher performance but needs strong cooling and a good power supply.
- example: a gpu with a higher board power will need extra power cables and a better PSU.

Thermal throttling

- what it is: automatic slowing down when a chip gets too hot.
- why it matters: heat can reduce frame rate even if your settings did not change.
- example: a laptop’s fps drops after ten minutes because it hits thermal limits.

Fan curve

- what it is: how fans speed up as temperatures rise.
- why it matters: you can tune the balance of noise and temperature.
- example: a more aggressive fan curve keeps the gpu cooler and prevents throttling.

Motherboard chipset and PCIe

- what it is: the board and the high speed lanes that connect cpu, gpu, and storage.
- why it matters: newer PCIe versions can help very fast gpus and SSDs.
- example: a PCIe 4.0 NVMe drive loads levels faster than an older SSD.

PSU

- what it is: the power supply.
- why it matters: stable power avoids random crashes and black screens.
- example: a 750 W quality PSU is safer for a strong mid range gpu than a cheap 500 W unit.

Storage: HDD vs SSD vs NVMe

- what it is: hard drives are slow, SSDs are faster, NVMe is the fastest common option.
- why it matters: faster storage loads maps faster and reduces texture pop in.
- example: moving a game from HDD to NVMe cuts load times to a few seconds.

Display and smoothness

Resolution

- what it is: the number of pixels on screen like 1920×1080 or 2560×1440 or 3840×2160.
- why it matters: more pixels look sharper but need more gpu power.
- example: 1440p looks cleaner than 1080p but can reduce frame rate.

Refresh rate

- what it is: how many times per second the monitor updates, measured in Hz.
- why it matters: higher refresh feels smoother and can reduce blur.
- example: 144 Hz looks smoother than 60 Hz in a fast shooter.

FPS

- what it is: frames per second. how many frames the gpu draws each second.
- why it matters: higher fps means smoother motion and lower input delay.
- example: 120 fps feels more responsive than 60 fps.

Frame time

- what it is: the time to draw each frame, measured in milliseconds.
- why it matters: even frame times feel smooth. uneven frame times feel stuttery.
- example: a game that sits at 10 ms per frame feels smooth. spikes to 40 ms feel like hiccups.

Input latency

- what it is: delay from your action to the result on screen.
- why it matters: lower latency helps aim and movement.
- example: a competitive player caps fps to match refresh and uses features that lower latency.

Screen tearing

- what it is: a visible split line when the gpu and monitor are out of sync.

- why it matters: it is distracting in fast motion.
- example: a quick camera turn shows a horizontal tear across the screen.

VSync, G Sync, FreeSync

- what it is: ways to sync gpu output with the monitor.
- why it matters: they reduce tearing and can improve smoothness.
- example: FreeSync adjusts refresh to match the fps window of the game.

Response time and overdrive

- what it is: how fast a pixel changes color.
- why it matters: slow response causes ghosting trails.
- example: setting overdrive too high creates inverse ghosting artifacts.

HDR and color gamut

- what it is: high dynamic range and wider color range.
- why it matters: better highlights and richer color if the game and display support it.
- example: a bright desert scene looks more realistic with HDR on a capable monitor.

Hardware Basics & CPU Architectures

CPU Hybrid Architecture (Intel)

- **What it is:** A CPU design, prominently used by Intel in its 12th generation and newer processors, that combines two different types of cores on one chip: Performance-cores (P-cores) and Efficient-cores (E-cores).
- **Why it matters:** P-cores handle demanding tasks like gaming, while E-cores take care of background processes like Discord or streaming software. This division of labor helps the game run more smoothly without interruption from background tasks.
- **Example:** You're gaming and streaming at the same time. The P-cores are focused on delivering high frame rates in your game, while the E-cores efficiently manage your streaming software, preventing stutter.

Chiplet Design (AMD)

- **What it is:** An approach used by AMD for its Ryzen™ CPUs where the processor is built from smaller, specialized modules called "chiplets" connected together, rather than one large, single "monolithic" chip.
- **Why it matters:** This design makes it more cost-effective to create CPUs with very high core counts and can lead to better manufacturing yields.
- **Example:** High-end AMD Ryzen™ 9 processors can offer 12 or 16 cores for powerful multitasking and content creation, which is made possible and more affordable by the chiplet design.

Graphics Techniques and Settings

AMD FidelityFX™ Super Resolution (FSR)

- **What it is:** AMD's upscaling technology. A key feature is that it's open-source, meaning it can work on a wide range of GPUs, including those from Nvidia and Intel.
- **Why it matters:** It provides a significant performance boost and is a great option for gamers regardless of which brand of graphics card they own. Because it's so widely compatible, more game developers are including it as an option.
- **Example:** You have an older Nvidia GTX 1660 card; you can enable FSR in a supported game to get a much-needed FPS boost to play smoothly.

NVIDIA® DLSS (Deep Learning Super Sampling)

- **What it is:** NVIDIA's AI-powered upscaling technology. It requires dedicated hardware on their RTX graphics cards called Tensor Cores to work.
- **Why it matters:** DLSS is highly regarded for producing high-quality images that can sometimes look even better than the native resolution, all while providing a major performance uplift.
- **Example:** In a graphically intense game like *Cyberpunk 2077*, turning on DLSS can be the difference between a choppy 40 FPS and a smooth 70 FPS on an RTX card.

Intel® XeSS (Xe Super Sampling)

- **What it is:** Intel's upscaling solution. Like DLSS, it uses AI to enhance image quality. It works best on Intel's Arc™ graphics cards but is also designed to work on a wide range of other GPUs.
- **Why it matters:** It gives gamers another powerful upscaling option, and its flexibility encourages wider adoption by game developers.
- **Example:** You're playing a new release on an Intel Arc™ GPU and enable XeSS to get a clean, high-resolution image with a higher frame rate.

Rasterization

- what it is: the classic way games draw triangles into pixels.
- why it matters: it is fast and used for most of the image.

Ray tracing

- what it is: simulating light rays for more realistic reflections, shadows, and lighting.
- why it matters: looks great but needs a lot of gpu power.
- example: reflections in a wet city street look accurate with ray tracing on.

Upscaling

- what it is: render at a lower resolution then scale up to your screen.
- why it matters: higher fps with similar sharpness if done well.
- example: render at 1440p, upscale to 4K to keep smooth play.

Frame generation

- what it is: the system creates extra frames between real frames.
- why it matters: increases apparent fps at the cost of some added latency and possible artifacts.
- example: a game goes from 70 fps to about 120 fps with frame generation enabled.

Anti aliasing

- what it is: methods that reduce jagged edges.
- why it matters: cleaner image at the cost of performance or blur.

- example: TAA reduces jaggies but can soften the image a bit, while MSAA is sharp but heavier.

Anisotropic filtering

- what it is: improves texture clarity at angles.
- why it matters: better ground and wall detail with tiny performance cost.
- example: road textures stay sharp as they recede into the distance.

Ambient occlusion

- what it is: soft shading in corners and where objects meet.
- why it matters: adds depth and realism.
- example: crates look grounded instead of floating.

Screen space reflections

- what it is: reflections made from what is already on screen.
- why it matters: cheaper than ray tracing but can miss things off screen.
- example: a mirror only reflects what the camera sees.

Level of detail

- what it is: the game reduces model detail at distance.
- why it matters: saves performance with minimal visual loss when tuned well.
- example: far away trees use simpler models.

Post processing

- what it is: effects applied after the main image is drawn.
- why it matters: bloom, motion blur, depth of field, film grain all change look and performance.
- example: turning motion blur off can make motion feel clearer in shooters.

Mesh shaders and culling

- what it is: modern ways to decide what to draw and what to skip.
- why it matters: better performance in complex scenes.

Tessellation

- what it is: adds more triangles to make surfaces smoother.
- why it matters: can improve detail but costs gpu time.

Software, Drivers, and Engines

Driver

- what it is: the software that lets your operating system and games talk to the gpu.
- why it matters: new drivers can improve performance and fix bugs, especially at launch.
- example: a day zero driver for a new game improves frame time stability.

Game engine

- what it is: the core software that runs the game like Unreal or Unity or proprietary engines.
- why it matters: engine choice affects performance, graphics features, and modding.

Graphics API

- what it is: the rules and functions a game uses to talk to the gpu like DirectX or Vulkan or OpenGL. Graphics APIs—sets of software instructions—that allow game developers to communicate with a computer's graphics card (GPU) to create and display 2D and 3D graphics on screen.
- DirectX is a proprietary Microsoft technology for Windows and Xbox, offering hardware-specific features but limited to Microsoft platforms.
- Vulkan is a modern, low-level, cross-platform API designed for high-performance, providing direct hardware control, improved multi-threading, and lower CPU overhead.
- OpenGL is an older, higher-level, cross-platform API that is widely supported but offers less direct control and may be less efficient than modern alternatives like Vulkan.
- why it matters: different APIs can perform differently on the same hardware.

Shader compilation and shader cache

- what it is: turning shader code into a form the gpu can run, then saving it for reuse.
- why it matters: compiling during gameplay can cause stutter, caching reduces it.
- example: a game builds its shader cache on first run, then plays smoother later.

Overlay

- what it is: on screen display for fps, temps, chat, or capture controls.
- why it matters: helpful but can conflict with anti cheat or each other.
- example: disable extra overlays during a ranked match to avoid issues.

Anti cheat

- what it is: software that tries to stop cheating.
- why it matters: can block some overlays and tools to keep matches fair.

First run experience

- what it is: the steps and defaults when you launch a game or driver app the first time.
- why it matters: smart defaults reduce support issues and make players happy.
- example: the app offers a recommended preset for your hardware at first launch.

NVIDIA® Reflex

- **What it is:** A technology that reduces system latency (the delay between your mouse click and the action appearing on screen). It requires game developers to integrate it directly into their games.
- **Why it matters:** In fast-paced competitive games, lower latency can give you a measurable advantage, making your aim feel more responsive. Reflex is highly effective because it optimizes the entire pipeline from the CPU to the GPU.
- **Example:** In a game like *Valorant* or *Apex Legends*, enabling NVIDIA® Reflex can make your shots feel like they're happening instantly, potentially helping you win more duels.

AMD Radeon™ Anti-Lag

- **What it is:** A driver-level feature from AMD that also aims to reduce input latency.

- **Why it matters:** It can make games feel more responsive, especially in situations where the GPU is the bottleneck. Since it's controlled through the driver, it can be enabled for a wide variety of games.
- **Example:** You enable Radeon™ Anti-Lag in the AMD software settings to get a snappier feel in your favorite shooter, even if the game doesn't have a built-in latency reduction option.

Network and online play

Ping

- what it is: how long it takes data to go to the server and back, measured in ms.
- why it matters: lower ping means faster hit registration and smoother play.

Jitter

- what it is: how much your ping varies over time.
- why it matters: even a low ping that jumps around can feel bad.

Packet loss

- what it is: data that never arrives.
- why it matters: causes rubber banding and missed shots.

Server tick rate

- what it is: how many times per second the server updates the game state.
- why it matters: higher tick rate can feel more responsive.

Bandwidth

- what it is: how much data your connection can carry per second.
- why it matters: matters more for downloads and streams than for basic gameplay.

NAT type

- what it is: how your router handles connections.
- why it matters: strict NAT can block voice chat and matchmaking.

Ethernet vs wifi

- what it is: wired vs wireless.
- why it matters: ethernet is usually more stable with lower latency.

Streaming and capture

Encoding and codec

- what it is: how video is compressed for streaming or recording, like H.264, HEVC, or AV1.
- why it matters: better codecs can look cleaner at the same bitrate.
- example: AV1 at 6 Mbps can look better than H.264 at the same bitrate.

Bitrate

- what it is: how much data per second for your video stream.
- why it matters: higher bitrate looks better but needs more upload speed.

CBR and VBR

- what it is: constant vs variable bitrate.
- why it matters: VBR can save bandwidth while keeping quality, CBR is more predictable.

Keyframe interval

- what it is: how often a full image is stored.
- why it matters: affects seek speed and stream stability.

Stream resolution and fps

- what it is: size and smoothness of your broadcast.
- why it matters: 1080p60 looks smooth if your upload can handle it.

Record buffer and instant replay

- what it is: keep a rolling buffer so you can save the last moments.
- why it matters: great for highlight clips without recording all the time.

Game options and performance tuning

Preset

- what it is: a group of settings like low, medium, high, ultra, or recommended.
- why it matters: a quick way to get good performance without tweaking everything.

Dynamic resolution

- what it is: the game adjusts render resolution on the fly to keep frame rate steady.
- why it matters: fewer frame time spikes during intense scenes.

CPU bound vs GPU bound

- what it is: which part is limiting performance.
- why it matters: if cpu bound, lower crowd density or physics. if gpu bound, lower shadows or resolution.

Thermal headroom

- what it is: how much temperature room you have before throttling.
- why it matters: better cooling keeps clocks higher.

Background processes

- what it is: other apps running while you game.
- why it matters: they can eat cpu, memory, and disk, causing stutter.

Exclusive fullscreen vs borderless windowed

- what it is: display modes.
- why it matters: exclusive fullscreen can reduce latency. borderless makes alt tab easy.

FOV

- what it is: field of view, how wide the camera sees.
- why it matters: higher FOV can help awareness but may cost performance.

Input and control

Polling rate

- what it is: how often your device reports its position.
- why it matters: higher rates can reduce input delay.
- example: a mouse at 1000 Hz feels more responsive than 125 Hz.

DPI and eDPI

- what it is: DPI is mouse sensitivity in hardware. eDPI is dpi times in game sensitivity.
- why it matters: helps you compare settings across games and mice.

Deadzone and acceleration

- what it is: deadzone is the stick movement that does nothing. acceleration speeds up movement based on how fast you move.
- why it matters: tuning these makes aim feel natural.

Advanced but useful

Resizable BAR

- **What it is:** An industry-standard technology that allows the CPU to access the entire pool of graphics card VRAM at once, rather than in small chunks. It's supported by modern AMD, Intel, and NVIDIA components.
- **Why it matters:** It removes a performance bottleneck, which can lead to a modest performance increase of up to 15% in certain games, especially at higher resolutions.
- **Example:** Enabling Resizable BAR in your computer's BIOS can give you a "free" performance boost, squeezing a few extra frames per second out of your existing hardware.

AMD Smart Access Memory (SAM)

- **What it is:** AMD's brand name for Resizable BAR technology. It was originally promoted for systems with an AMD CPU and AMD GPU working together.
- **Why it matters:** Like Resizable BAR, it can provide a noticeable performance uplift in supported games. The performance difference between SAM and a standard Resizable BAR implementation is generally minimal.

- **Example:** If you have a PC with a Ryzen™ 5000 series (or newer) CPU and a Radeon™ RX 6000 series (or newer) GPU, enabling SAM is a quick way to potentially boost your gaming performance.

Display scaling

- what it is: resize the image to fit your screen.
- why it matters: affects sharpness and input delay depending on the method.

HBM and GDDR

- what it is: types of gpu memory. HBM is very fast stacked memory. GDDR is common on gaming cards.
- why it matters: memory type and speed affect bandwidth.

L2 cache and large gpu cache

- what it is: memory inside the gpu that reduces trips to VRAM.
- why it matters: helps frame time stability in texture heavy scenes.

Power limit and undervolting

- what it is: controlling how much power the gpu can draw or lowering voltage for the same clock.
- why it matters: can lower temperatures and noise while keeping performance.

Shader

- what it is: small programs that tell the gpu how to draw pixels, vertices, and effects.
- why it matters: efficient shaders improve performance and visuals.

Tiled or deferred rendering

Tiled/deferred rendering is a graphics processing technique that optimizes rendering by dividing the screen into small "tiles" and processing each one separately. Instead of calculating lighting for every geometry object in the whole scene at once, rendering is "deferred" until after the geometry for each tile has been determined, allowing for parallel processing and reduced memory usage, especially for mobile and low-power devices.

- **Tiled Rendering:**
  - The screen is divided into a grid of smaller regions called "tiles".

  - The GPU processes each tile independently.
  - This reduces the amount of memory and bandwidth needed compared to drawing the entire frame at once.
  - It's particularly useful for devices with limited power and memory, such as handheld devices and mobile phones.
- **Deferred Rendering:**
  - Lighting calculations are "deferred" or "postponed" until after the scene's geometry has been processed for each tile.
  - This is a screen-space lighting technique, meaning it operates on the image space rather than the 3D geometry.
  - By deferring lighting, the GPU can determine which parts of the geometry are relevant to each tile before performing lighting calculations, which is more efficient.

How they work together:

- **Tiled-Based Deferred Rendering (TBDR)**: is a combined technique where the GPU breaks the screen into tiles and then defers the lighting calculations for each tile.
- This allows the GPU to efficiently process many tiles in parallel, optimizing for both performance and power efficiency.
- The goal is to improve efficiency by determining all the geometry for a tile first, then performing the computationally expensive lighting calculations on that tile, rather than scattering the work across the entire frame.

GeForce Experience vs. AMD Software: Adrenalin Edition

NVIDIA® GeForce Experience™ (and the new NVIDIA® App)

- **What it is:** A companion application for NVIDIA GPUs that provides game-ready drivers, automatic game optimization, and features like ShadowPlay for recording gameplay highlights.
- **Why it matters:** It's a convenient, all-in-one tool for keeping drivers updated and easily capturing your best gaming moments. The new NVIDIA® App aims to streamline this experience further.

- **Example:** Using the instant replay feature (ShadowPlay) to save a clip of an amazing shot you just made without having to record your entire gaming session.

AMD Software: Adrenalin Edition™

- **What it is:** The all-in-one software suite for AMD Radeon™ graphics cards, combining driver updates, performance monitoring and tuning, recording and streaming features, and game-specific settings.
- **Why it matters:** It's praised for integrating a vast number of features into a single, cohesive application, from overclocking to latency reduction settings.
- **Example:** Opening AMD Software to enable Radeon™ Anti-Lag for a specific game, check your GPU temperature, and update your driver all from the same interface.

---

quick examples that tie it together

- you want smoother play in a shooter on a mid range gpu
  set the preset to high, use upscaling quality mode, cap fps to your monitor refresh, and enable the variable refresh feature your monitor supports. this balances gpu load and reduces tearing.
- you see stutter in a new open world game even on a strong gpu
  update the driver, let the game pre build shaders, close background apps, and lower only the two heavy settings for your gpu like shadows and volumetrics. check frame time graph rather than only average fps.
- your stream looks blocky
  raise bitrate if your upload allows, switch to a better codec if your viewers support it, and use a performance safe capture preset so your game stays smooth.