

Kamera-Frustum

Englisch: Camera Frustum

VFX

Der Sichtbereich der Kamera im 3D-Raum — definiert durch Nah- und Fernebene, Öffnungswinkel. Alles außerhalb wird nicht gerendert.

Im 3D-Raum existiert eine unsichtbare pyramidenförmige Zone — der Frustum. Alles, was darin liegt, wird gerendert; alles davor und dahinter verschwindet. Als VFX-Supervisor oder im Motion-Design-Setup muss dieser Bereich kontrolliert werden, sonst geht entweder Performance verloren oder wichtige Objekte verschwinden mitten im Shot. Der Frustum wird durch vier Parameter definiert: Near Plane (Naheebene), Far Plane (Fernebene), Field of View (horizontaler oder vertikaler Öffnungswinkel) und das Seitenverhältnis des Ausgabeformats. Im praktischen Workflow bedeutet das: Bei einer Near Plane von 0,1 Einheiten wird nichts gerendert, das näher an der virtuellen Kamera liegt. Eine Far Plane von 10.000 Einheiten kann dagegen zu unnötigen Rechenprozessen führen, wenn die Szene ohnehin bei 5.000 endet. Jede Software — Maya, Cinema 4D, Blender, Houdini — visualisiert diesen Kegel unterschiedlich, aber das Prinzip bleibt gleich. Im Motion-Graphics-Kontext wird das kritisch beim Einsatz von volumetrischen Effekten, Partikel-Systemen oder Depth-of-Field-Berechnungen. Ein Particle-Emitter außerhalb des Frustums existiert in der Output-Sequenz nicht — trotz korrekter Simulation. Häufig passiert das unbewusst bei großen Kamera-Pullouts oder Drohnen-Shots mit weitwinkeligen Linsen. Sitzt die Near-Plane zu weit hinten, fällt die Nahaufnahme weg. Umgekehrt: Liegt die Far Plane zu nah, clippen fernere Geometrien ab — im Compositing sichtbar als unsaubere Kanten. Best Practice am Set oder in der VFX-Suite: Frustum-Visualisierung aktivieren. Jede 3D-Engine zeigt den Kegel-Umriss auf Anfrage. Damit lässt sich sofort kontrollieren, ob Effekte, Beleuchtung oder Geometry verloren gehen. Bei VFX-Heavy-Shots — besonders bei Kamera-Movements durch komplexe Szenen — gehört der Frustum-Bereich auch in Relation zur Scene-Scale geprüft. Wird im Team in Zentimetern gearbeitet, die Kamera aber in Metern kalibriert, entstehen schnell Clipping-Fehler. Ein zusätzlicher Schritt vor dem Render: Frustum-Check in den Render-Settings und die Parameter im Asset-Export dokumentieren. Das spart Re-Renders. Aktuelles In der Praxis wird das Kamera-Frustum vor allem für das sogenannte Frustum Culling genutzt: Rendering-Engines wie Houdini oder Unity prüfen automatisch, welche Objekte innerhalb des Sichtkegels liegen, und schließen alles andere von der Berechnung aus. Das spart Rechenzeit, besonders bei komplexen Szenen mit vielen Assets, etwa Crowd-Simulationen. Tutorials und Foren-Diskussionen zeigen, dass Artists zunehmend eigene Frustum-Setups bauen, um Culling gezielt zu steuern oder Volumen zu definieren, die das Rendering blockieren.