

Frustum Culling

Englisch: View Frustum Culling

VFX

Rendering-Technik, die Geometrie außerhalb des Kamera-Sichtfeldes ausblendet — spart Rechenleistung und beschleunigt die Engine. Standard bei Echtzeit-VFX und Spielmotoren.

Die Kamera sieht nur einen bestimmten Bereich der Szene — alles dahinter, davor oder seitlich außerhalb des Sichtfeldes ist für den Frame irrelevant. Frustum Culling nutzt diese Realität: Die Engine berechnet zunächst den pyramidenförmigen Sichtbereich (den "Frustum"), der vom Kamera-Objektiv definiert wird, und wirft dann alle Geometrie raus, die darin nicht liegt. Was nicht gerendert wird, kostet auch keine Rechenzeit. Praktisch funktioniert das so — jedes 3D-Objekt oder Mesh bekommt eine unsichtbare Bounding Box. Die Engine prüft beim Rendering-Durchgang, ob diese Box mit dem Frustum überlappt. Liegt sie komplett außerhalb, wird das Objekt von der Rendering-Pipeline entfernt, bevor es überhaupt zur GPU kommt. Das spart nicht nur Vertex-Durchsätze, sondern auch Shader-Aufrufe und Texture-Bandbreite. Bei komplexen VFX-Szenen mit hunderten Partikelsystemen, Lichtern oder deformierten Meshes ist das der Unterschied zwischen stabilen 24fps und Echtzeit-Chaos. Im VFX-Workflow ist das ein alltägliches Prinzip: Ein Destruction-Shot mit flying debris füllt die ganze Szene mit Trümmern, aber nur die vordere Hälfte ist im Kamera-Frame relevant. Statt alle Partikel-Kollisionen zu berechnen, lässt sich via Frustum Culling die Simulation auf die sichtbaren Trümmer beschränken. Ähnlich bei Crowd-Szenen — Hunderte Character-Rigs können in der Szene liegen; gerendert und animiert werden nur die im Sichtbereich. Auch bei Lichtern gilt: Zehn Lichter außerhalb des Frustums brauchen nicht in die Shader-Berechnung einzugehen. Entscheidend ist die Grenze zwischen Near- und Far-Plane der Kamera. Zu nah gezogen, wird versehentlich sichtbare Geometry gecullt; zu weit, geht der Effizienz-Gewinn verloren. Manche Engines erlauben feinere Kontrolle über Culling-Distanzen pro Objekt — gerade für Parallax-Backgrounds oder sehr ferne Details hilfreich. Auch der Culling-Overhead selbst ist real: Bei sehr einfachen Szenen kann Frustum Culling langsamer sein als einfach alles zu rendern. Deshalb wird es vor allem in komplexeren Szenen aktiv oder kann manuell gesteuert werden. Verwandt mit anderen Optimierungs-Techniken wie Occlusion Culling (was von anderen Objekten verdeckt wird) und LOD-Systemen (weiter entfernte Objekte mit weniger Details). Frustum Culling ist aber die Basis — es ist Standard in allen modernen Game-Engines und wird bei VFX-Rendering automatisch angewandt.