

Z-Puffer

Englisch: Z-buffer

VFX

Speicherbereich in der 3D-Engine, der die Tiefenwerte jeden Pixels speichert — entscheidet, welche Geometrie vorne liegt. Ohne Z-Buffer keine korrekte Verdeckung.

Im Compositing wirken 3D-Render manchmal durcheinander — Objekte überlagern sich falsch, vorne liegt, was hinten sein sollte. Der Z-Puffer arbeitet dagegen: Er speichert für jeden Pixel die Tiefenkoordinate — den Abstand zur Kamera — und entscheidet damit in Echtzeit, welche Geometrie sichtbar bleibt und welche verdeckt wird. Ohne diesen Speicherbereich würde die 3D-Engine nicht wissen, ob ein Würfel vor oder hinter einem anderen liegt. In der Praxis funktioniert das so: Während die GPU die Szene rendert, speichert sie parallel zu jedem Pixel nicht nur Farbe, sondern auch seinen Z-Wert — die Tiefe. Trifft ein neuer Pixel auf dieselbe Position, vergleicht die Engine seine Z-Tiefe mit dem gespeicherten Wert. Nur wenn der neue Pixel näher zur Kamera liegt, wird er geschrieben. Das ist der Depth-Test, und er läuft Millionen Mal pro Frame ab. Ohne Z-Puffer müsste jedes Mal manuell sortiert werden, welche Layer oben liegt. Am Set der VFX-Produktion wird der Z-Puffer als Depth-Pass oder Z-Depth-Pass benötigt. Das ist eine separate Render-Layer, die statt der Farben die Tiefenwerte visualisiert — hell für nah, dunkel für fern. Im Compositing lässt sich damit Tiefenunschärfe realisieren, Volumetrics realistisch einblenden oder Keying präziser machen. Viele Renderer speichern Z-Puffer mittlerweile als 32-Bit- oder 16-Bit-Channel — je höher, desto subtiler die Abstufungen, desto weniger Banding im Bokeh. Ein Stolperfallen-Punkt: Z-Fighting entsteht, wenn zwei Geometrien exakt dieselbe Tiefe haben. Dann flimmert der Z-Puffer zwischen den beiden Objekten hin und her — ein nerviges Glitchen. Vermeidbar durch kleine Offset-Werte oder korrektes Polygon-Arrangement. Bei Volumetrics oder transparenten Objekten ist außerdem zu beachten, dass der Z-Puffer nur mit dem klassischen Depth-Test funktioniert — manche Effekte brauchen Order-Independent Transparency (OIT), einen anderen Algorithmus, der Transparent-Probleme eleganter löst.