Grafikeffekte entschlüsselt

DirectX 10, High Dynamic Range Rendering, Parallax Occlusion Mapping – wer soll da noch durchsteigen? Wir erklären die komplizierte Technik.

DirectX 10

- Was es bringt: **im Idealfall schnellere Berechnung von Grafikeffekten**
- Wer es kann: **Geforce 8 und Radeon HD**
- Was es kostet: **derzeit noch sehr viel Leistung**

DirectX ist eine Sammlung von Programmierschnittstellen für Grafik, Sound und Multimedia.

Bereits seit über 11 Jahren gibt es DirectX. Musste Microsoft am Anfang noch um die Unterstützung der Spieleentwickler bangen, hat sich mit der Version 3.0 der Großteil der Industrie auf diesen Standard geeinigt. Die ersten Spiele, die Direct3D von DirectX nutzten, waren 1996 **Tomb Raider 2** und **Diablo**. Der Nachteil: Die Titel liefen nicht mehr unter dem bis dahin weit verbreiteten DOS (Betriebssystem ohne grafische Oberfläche) – Windows wurde so zum Betriebssystem Nummer eins für Spieler. Ein weiterer, wichtiger Meilenstein war DirectX 8.0 im Jahr 2000. Erstmals wurden Grafikprozessoren programmierbar, allerdings noch mit erheblichen Einschränkungen. Trotzdem waren die Entwickler wesentlich freier bei der Gestaltung von Effekten als je zuvor.

Grafikkarten wie die Geforce-8-Serie von Nvidia oder ATIs Radeon HD unterstützen die neueste Variante, DirectX 10 oder richtiger Direct3D 10. Wie bei DirectX 3.0 müssen Spieler aber zwangsweise auf ein neues Betriebssystem wechseln, wenn Sie von der aktuellen Schnittstelle profitieren wollen – diesmal auf Windows Vista.

Direkte optische Vorteile hat die neue Generation gegenüber DirectX 9 kaum, dafür bietet sie den Spieleentwicklern neue kreative Möglichkeiten, zum Beispiel durch den Geometry Shader (siehe Shader). Zudem wurden die unterschiedlichen Pixel-Prozessoren vereinheitlicht. Einfach gesagt: Fallen in einer Szene hauptsächlich Pixelarbeiten wie Texturen oder Rauch an, liegen bei der Direct-X-9-Hardware die Geometrieeinheiten brach. Unter DirectX 10 packen je nach Lastverteilung alle Einheiten mit an. Die Folge sind entweder spektakulärere Effekte oder einfach eine höhere Bildwiederholrate. Bisher unterstützen aber erst wenige Titel DirectX 10. HW

Das erste Spiel, das von Grund auf für **DirectX 10** entwickelt wird, ist Crysis.

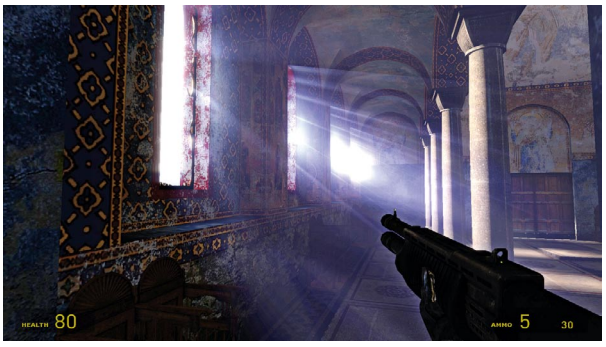
High Dynamic Range Rendering

- Was es bringt: **realistischere Beleuchtungseffekte in Spielen**
- Wer es kann: **Geforce 6, 7, 8 sowie Radeon X1000, HD**
- Was es kostet: **viel bis sehr viel Leistung**

Seit Far Cry 1.3 und Splinter Cell 3 blendet uns Licht: dank High Dynamic Range Rendering.

High Dynamic Range Rendering oder kurz HDR bietet ein wesentlich größeres Farbspektrum und feinere Abstufungen bei Farbübergängen. Durch eine Vervierfachung der Farbinformationen ermöglicht HDR beispielsweise atemberaubende Blendeffekte, die teilweise die Konturen von Baumwipfeln oder Fenstern (siehe Screenshot) überstrahlen. Zudem erlaubt High Dynamic Range Rendering bei sehr hellen Lichtquellen realitätsgetreue Überbelichtungseffekte. Fahren Sie beispielsweise in einem Rennspiel aus einem dunklen Tunnel ins helle Sonnenlicht, werden Sie kurzzeitig geblendet und sehen erst mal nichts, denn wie im echten Leben müssen sich die Augen an den Helligkeitsunterschied gewöhnen. Um HDR auszureizen, ist das Shader Model 3.0 von DirectX 9.0c Pflicht. Nvidia unterstützt das seit der Geforce 6-Familie, ATI zog mit seiner X1000er-Grafikchipgeneration später nach. Konnte Nvidia bei den ersten Karten der Geforce-6-Serie noch kein HDR und Kantenglättung gleichzeitig einsetzen, unterstützen mittlerweile alle neueren Karten von ATI (Radeon X1000, HD 2000) und Nvidia (Geforce 7 und 8) die beiden Effekte simultan. HW

Ohne High Dynamic Range Rendering scheint das Licht unnatürlich regelmäßig auf alle Objekte. So wirken beispielsweise die Säulen durch die gleichmäßige Beleuchtung platt.



Aktivieren Sie HDR, verbreitert sich das darstellbare Lichtspektrum. Das Sonnenlicht überstrahlt die Konturen der Fenster und beleuchtet die Säulen unterschiedlich.

Wie hier in F.E.A.R. ermöglichen Shader auch schicke **Raumkrümmungseffekte**.

Shader

- Was es bringt: **schnellere und umfangreichere Berechnung von Effekten**
- Wer es kann: **alle Grafikkarten**
- Was es kostet: **je nach Effekt keine bis sehr viel Leistung**

Shader sind die Heinzelmännchen der Grafikkarte; man sieht sie nicht, aber ihre Arbeit erledigen sie mit Bravour.

Shader sind an und für sich keine Grafikeffekte, sondern kleine Recheneinheiten, die auf der Grafikkarte laufen und nach Belieben des Programmierers Pixel und Polygone verändern können. Dabei unterscheidet man zwischen drei Varianten: Pixel Shader, Vertex Shader und Geometry Shader. Pixel Shader sorgen etwa für Wasserspiegelungen, modelierte Oberflächen oder Beleuchtung. Parallax Occlusion Mapping zum Beispiel lässt eine platte Textur dreidimensional wirken (siehe Screenshot). Dass die Textur aber in Wirklichkeit flach und zweidimensional bleibt, erkennen Sie leicht im hinteren Bereich des Backsteinpflasters.

Vertex Shader kümmern sich um die Polygone, und damit auch um Animationen oder Deformationen – schön zu sehen zum Beispiel am Schadensmodell der Fahrzeuge in **Need for Speed: Pro Street**. Selbst die Kratzer im Lack werden von den kleinen Recheneinheiten dargestellt. Ein weiteres Anwendungsgebiet ist Geometry Instancing. Das sorgt für Objektvielfalt ohne Leistungseinbußen, indem es etwa die Einheiten in **Herr der Ringe: Schlacht um Mittelerde 2** beliebig oft neu erstellt und zeitlich versetzt animiert, sodass die riesigen Heere der Orks und Elfen organisch wirken und nicht, wie noch vor wenigen Jahren, komplett synchron marschieren und kämpfen.

Neu im Shader-Team sind seit DirectX 10 die Geometry Shader. Diese Spezialisten können die von den Vertex-Shadern ausgegebenen Polygondaten weiterverarbeiten und so beispielsweise zusätzliche Polygone zu einer Szene hinzufügen. Denkbar wäre hier ein Rollenspiel, in dem Sie als Magier ganze Gebirge erschaffen und untergehen lassen, um Ihren Feinden eine Falle zu stellen. Leider nutzt bisher noch kein Spiel dies konsequent. HW

Durch **Parallax Occlusion Mapping** erscheinen zweidimensionale Texturen wie dieser gepflasterte Boden räumlich – nur ein Anwendungsgebiet von Shadern.

Schönere Objekte

- Was es bringt: **mehr Details erhöhen die Glaubwürdigkeit**
- Wer es kann: **alle Grafikkarten**
- Was es kostet: **wenig bis viel Leistung**

In den letzten Jahren stieg nicht nur die Anzahl der Objekte in Spielen rapide an, sondern auch deren Detailfülle.

Egal ob Gesichter, Bäume oder Fässer: Objekte in Spielen werden stets detaillierter. Immer schnellere Grafikkarten ermöglichen sprunghaft steigende Polygonzahlen. Wo früher wenige hundert Dreiecke ausreichen mussten, bestehen Charakter heute aus mehreren tausend Polygonen. Wo früher eine Hand nicht mal Finger hatte, erkennen wir heute dank moderner Texturverfahren feinste Hautdetails wie Poren oder Leberflecken. Genauso schnell entwickelten sich die Gesichter. In **Half-Life** (1998) versuchte ein kaum beweglicher, fast rechteckiger Mund die Illusion von sprechenden Charakteren zu transportieren. In **Half-Life 2** (2004) redeten der G-Man, Alyx & Co durch viele simulierte Muskeln schon lippensynchron. Cryteks CryEngine 2 verpasst der Gesichtshaut mit »Subsurface Scattering« zusätzlich Tiefe – die Hauttextur lässt offenbar darunterliegende Hautschichten durchscheinen. Das Gleiche funktioniert in **Crysis** auch mit Blättern ganz hervorragend. Sie lassen nun das Licht stark gefiltert hindurch und erzeugen einen gedämpften, weichen Schatten – ganz wie in echten Wäldern. HW

Vom **Polygonhaufen** (Half-Life) zum **Menschen** (Half-Life 2).

Tiefen- und Bewegungsunschärfe

- Was es bringt: **besseres Gefühl für Geschwindigkeit oder Entfernung**
- Wer es kann: **alle Grafikkarten**
- Was es kostet: **kaum bis wenig Leistung**

Um der Realität so nahe wie möglich zu kommen, simulieren die Programmierer das Fokusverhalten unserer Augen – ähnlich wie in Kinofilmen.

Schnell fahrende Autos verwischen in der Realität, weil unser Auge die hohe Geschwindigkeit nicht wahrnehmen kann. Am PC muss dieser Effekt nachträglich eingefügt werden. Vereinfacht gesagt, erstellt die Engine dazu Kopien der zu verwischenden Objekte und verschiebt diese in Abhängigkeit zur Geschwindigkeit entgegen der Bewegungsrichtung – Bewegungsunschärfe entsteht (»Motion Blurring«). Einen groben Eindruck des Verfahrens vermitteln die Spuren des Mauszeigers unter Windows. Besonders intensiv verwendet beispielsweise **Just Cause** Bewegungsunschärfe. Bei hohem Tempo verschwimmt die Umwelt stärker, als das in der Realität der Fall wäre.

Tiefenunschärfe (»Depth of Field«) imitiert unser Sehvermögen insofern, dass Gegenstände im Fokus gestochen scharf gezeichnet werden, und die umliegenden Objekte unscharf. **Crysis** demonstriert das etwa beim Einsatz des Scharfschützengewehres. HW

Unschärfeeffekte gaukeln dem Auge Bewegung vor.

DirectX-9-Karten mit Shader Model 2.0 wie die Radeon X800 unterstützen Spieleentwickler nur noch spärlich.

Licht & Schatten

- Was es bringt: **dichtere Atmosphäre**
- Wer es kann: **je nach Schattentyp unterschiedliche Grafikkarten**
- Was es kostet: **viel bis sehr viel Leistung**

Schatten erzeugen Atmosphäre und erhöhen den Realismus – auch wenn die Doom-3-Gegner wenig real aussehen.

Was wäre Doom 3 ohne Schattenspiele? Licht und Schatten erzeugen zwar viel Atmosphäre, fressen aber auch viel Leistung.

Raffinierte Schattenspiele regen unsere Fantasie an und können wesentlich gruseliger sein als plumpe Zombies. Am Anfang der Computerspiele bestand ein Schatten bestenfalls aus einer schwarzen Textur, die unter einen Gegenstand gelegt wurde. Beim »Stencil Shadows«-Verfahren der **Doom 3**-Engine entsprachen die Schatten erstmals genau dem Objekt, das sie wirft. Allerdings kranken diese »Stencil Shadows« an unrealistischen, harten Kanten. Unter anderem, weil die Lichtreflexion durch die Umgebung fehlt. Als eines der ersten Spiele trieb **F.E.A.R.** mit weichen Schatten die Hardware an ihre Grenzen.

Die sogenannte globale Beleuchtung wird immer wichtiger: Die Engine berechnet nicht nur den Verlauf der Sonne, die Reflexion der Lichtstrahlen auf Objekten und zusätzliche Lichtquellen, sondern auch die entsprechenden Schatten – egal ob Haus, Baum oder Wolken. Obwohl das Unmengen an Leistung frisst, werden in Zukunft immer mehr Spiele auf diese Art der Schattenberechnung zurückgreifen. Crysis fängt noch dieses Jahr damit an. HW

Wassereffekte

- Was es bringt: **glaubwürdigere Spielwelt**
- Wer es kann: **alle Grafikkarten; qualitative Unterschiede**
- Was es kostet: **wenig bis sehr viel Leistung**

An der naturgetreuen Darstellung von Wasser sind schon viele Grafikengines gescheitert.

Bereits die ersten 2D-Spiele versuchten Wasser abzubilden, damals ganz simpel als blaue Texturtapete. Gischt und Wellenbewegungen wurden mit einfachen, sich bewegenden Pixeln (Sprites) simuliert. Selbst in den ersten 3D-Spielen (Anfang der 90er-Jahre) war Wasser nur eine Textur, die über ein Becken gelegt wurde. Der Spieler sank bis zu den Knien ein, so sah Wasser damals aus. Als eines der ersten Spiele versuchte **Tomb Raider**, Wasser nachvollziehbar darzustellen – und schaffte es; selbst Tauchen war kein Problem. Eine blau eingefärbte Umgebung plus einige Lichteffekte, fertig war die Illusion.

Einen weiteren großen Schritt hin zu scheinbar echtem Wasser machte **Morrowind**, indem es das kühle Nass über einen Shader berechnet. Dadurch zeigte Wasser erstmals so etwas wie Wellenbewegungen. Dennoch wirkt das Wasser nicht wirklich flüssig, sondern eher zäh wie Gelee. Dass Wasser mehr kann als nur fließen, zeigte **Splinter Cell: Chaos Theory.** Die Entwickler verpassten dem Wasser »leitende« Fähigkeiten, wodurch wir in Pfützen stehende Gegner durch Elektroschocks bequem grillen konnten. HW

Selbst Bioshock scheitert an glaubwürdiger **Wassersimulation**.

Sichtweite

- Was es bringt: **weniger Rechenarbeit dank weniger Details**
- Wer es kann: **alle Grafikkarten**
- Was es kostet: **je nach Einsatz Leistungsgewinn oder -Verlust**

Je weiter ein Objekt entfernt ist, desto weniger Details erkennen wir – Spiele nutzen dies für bessere Performance.

Eine Wand in der Ferne wirkt zunächst einfach nur grau; je näher Sie ihr kommen, desto mehr Details wie abbröckelnden Putz oder darunterliegende Steine erkennen Sie. Die Details waren zwar auch vorher da, verwischen aber mit zunehmenden Abstand. Diesen Effekt machen sich viele Spiele zunutze, um Rechenzeit zu sparen. Je nach Einstellung reduzieren sie mit abnehmender Sichtweite die Objektdetails, den sogenannten »Level of Details«. Oft streicht die Engine dann gleich ganze Gebirge oder Städte und vertuscht die mangelnde Fernsicht mit einer hässlichen Nebelsuppe. **Oblivion** ersäuft bei Auswahl der geringsten Sichtweite beispielsweise geradezu im Nebel. Aber auch hässliche Texturen kommen durch ein Abnehmen der Details zum Vorschein. So sinkt nicht nur der Detailgrad von Objekten in der Ferne, sondern auch Wiesen und Felder werden nur noch durch Farbverläufe dargestellt. Auch **Gothic 3** verzichtet nicht auf diese Tricks – dafür spart der unschöne Effekt deutliche Rechenzeit und erhöht die Leistung auf alten PCs enorm. HW

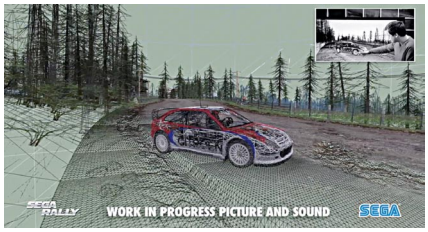
Ein **übertriebener Nebeleffekt** kaschiert in vielen Spielen die fehlenden Objekte in der Ferne – auch in Oblivion.

Oberflächen

- Was es bringt: **plastische Spielumgebungen**
- Wer es kann: **je nach Technik alle Grafikkarten oder erst neue Modelle**
- Was es kostet: **wenig bis viel Leistung**

Erst Texturen verpassen Polygonen eine realistische Oberfläche. Dabei sorgen sie nicht nur für einfache Muster, sondern auch für echte Dreidimensionalität.

Texturen kleben wie eine Tapete auf den flachen Polygonen und verleihen ihnen so eine annähernd realistische Oberfläche. In der Realität sind ebene Flächen aber eher selten, Mauern oder Fußböden haben immer eine Maserung wie Ziegel oder Beton. Diese Unebenheiten versuchen die Spieledesigner mit Tricks darzustellen. Einer der einfachsten Methoden ist das Bump Mapping. Dazu erstellt der Grafiker ein detailliertes und ein grobes Modell, beispielsweise von einer Straße. Da der Detailgrad der aufwändigen Textur aktuelle Grafikkarten überfordert, werden die Unterschiede zwischen beiden Modellen berechnet. Aus dieser Differenz wird eine Textur gefertigt, die exakt das abbildet, was dem einfacheren 3D-Modell an Details fehlt. Nur eben als simpel zu berechnende, flache 2D-Tapete. Die wird nun auf das abgespeckte 3D-Spielmodell aufgeklebt und peppt es optisch auf. Allerdings lässt sich diese Methode einfach entlarven, wenn Sie aus einem flachen Winkel auf die Straße schauen. Der neueste Trick der Entwickler, Displacement Mapping, fügt unserer Straße vereinzelte, echte Polygone hinzu. Dadurch wirkt die Straße nicht mehr nur uneben – sie ist es auch. HW



In Sega Rally buddeln die Autos **Furchen in den Boden**.

Physik und Partikel

- Was es bringt: **schönere Explosionen und nachvollziehbare Reaktionen von Objekten**
- Wer es kann: **alle PCs, manche Engines nur mit Zusatzkarte**
- Was es kostet: **viel bis sehr viel Leistung**

Neben immer feinerer Grafik spielt künftig besonders die Physik eine große Rolle, wenn Spiele eine in sich schlüssige Welt aufbauen wollen.

Damit Computerspiele realistisch wirken, müssen die Entwickler die echte Welt so detailliert wie möglich abbilden. Dazu gehören nicht nur bessere Grafikeffekte, sondern auch die nachvollziehbare Darstellung von physikalischen Gesetzmäßigkeiten, wie wir sie auch im Alltag erleben. So hat ein Porzellanteller nach dem Aufprall auf dem Boden in einer ganz bestimmten Art und Weise zu zerspringen, die Scherben müssen in einem gewissen Radius zu finden sein, und das typische Geräusch des berstenden Geschirrs soll bitte innerhalb der gewohnten Zeit abklingen. Denn sonst stellt sich ein Gefühl der Unwirklichkeit ein, das, wenn es nicht als Horroreffekt gezielt eingesetzt wird, die mühsam aufgebaute virtuelle Wirklichkeit mit einem Schlag zunichte macht. Bei der Zahl von Objekten, die mittlerweile durch die Levels kegeln, ist es eine gewaltige Aufgabe, die dargestellte Welt auch nur einigermaßen glaubwürdig in den Fugen zu halten. Wenn wir etwa im Spiel eine Tonne umtreten, erwarten wir mittlerweile, dass sie korrekt fällt – eventuell sogar weiterrollt und andere Gegenstände mit sich reißt, die dann wiederum nachvollziehbar reagieren. Tut sie das nicht, verpufft die Illusion der Wirklichkeit. Schon im Puzzlespiel **The Incredible Machine** (1992) unter DOS waren erste Ansätze korrekter Physik zu erkennen. Aber erst **Half-Life 2** verhalf ihr mit seinen Physikrätseln, die der Spieler per Gravity Gun löst, zum Durchbruch. Was in der Realität durch Kräfte wie die Erdanziehung selbstverständlich ist, müssen Spiele aufwändig berechnen. Schon längst gibt es neben den Grafikengines auch Physikengines, die sich nur um die Simulation nachvollziehbarer Wechselwirkungen kümmern. So will der Hersteller Ageia mit seiner PhysX-Karte solche Effekte berechnen und seine Technologie zu einem neuen Standard machen – bislang mit mäßigem Erfolg. Einen anderen Weg geht die Havok-Engine (**Half-Life 2**), die Physik von der CPU berechnen lässt.

In World in Conflict reagiert Rauch auf **Umwelteinflüsse** wie Wind oder Regen.

In **World in Conflict,** das ebenfalls auf Havok setzt, sorgen Partikeleffekte für dichte Staubwolken. In den meisten älteren Spielen bestehen Rauch oder Staub nur aus zweidimensionalen Bitmaps. Das wirkt zwar auf den ersten Blick realistisch, aber physikalische Effekte haben keinen Einfluss auf die optische Darstellung. Erst, seitdem Partikel wirklich dreidimensional dargestellt werden, spielt auch die Physik eine Rolle. In **Battlefield 2** ist der Rauch noch zweidimensional; besonders störend fällt dies auf, wenn er scharfe Kanten an Hindernissen bildet. **Crysis** und **World in Conflict** verwenden Soft Partikel, und die verhalten sich organisch, fließen also um Gegenstände herum. Nur so kann der Rauch vom Wind verwirbelt werden. Beeinflussen in **World in Conflict** beispielsweise die Rotorblätter der Helikopter den Rauch, treibt in **Far Cry 2** der Wind den Staub oder das Feuer über die Steppe. Allerdings werden Soft Partikel bisher nur unter DirectX 10 verwendet – laut Massive, dem Entwicklerstudio von **World in Conflict**, kosten sie unter DirectX 9 zu viel Leistung. HW

In Cellfactor unterliegen alle Objekte **physikalischen Gesetzen**.

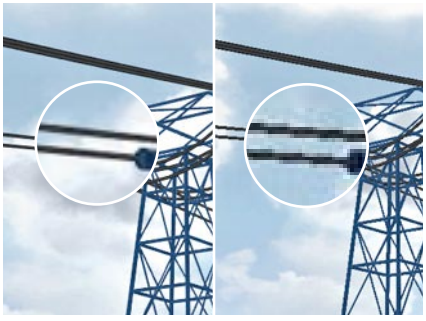
Kantenglättung und Texturfilter

- Was es bringt: **Kanten oder Texturen flimmern nicht mehr**
- Wer es kann: **alle Grafikkarten**
- Was es kostet: **je nach Qualitätseinstellung wenig bis viel Leistung**

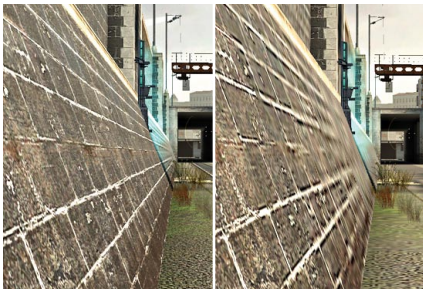
Kantenglättung und anisotrope Filterung sorgen für saubere Kanten und scharfe Texturen.

Der kleinste darzustellende Punkt eines Computerbildes ist ein Pixel. Besonders gut erkennt man dies bei einem TFT-Monitor, dort entspricht ein Pixel genau einem Bildpunkt. Pixel sind allerdings immer einfarbig, und genau das ist das Problem. Bilden nun beispielsweise mehrere schwarze Pixel eine schräge Polygonkante auf weißem Grund, wirkt das für den Betrachter wie eine Treppe. Bei besonders feinen Details wie Stromkabeln oder Zäunen fehlen aus der Entfernung manchmal sogar Teilstücke. Um Schrägen auf dem Monitor besser darzustellen, fügt Kantenglättung (Anti Aliasing, kurz AA) an den Treppchen sogenannte Subpixel hinzu, die es den Farbwerten der Kante und des Hintergrunds hinzumischt. Transparency AA (TSAA) funktioniert auch mit ganzen Maschendrahtzäunen oder ähnlichen Konstrukten, die eigentlich nur aus durchsichtigen Texturen bestehen.

Der anisotrope Texturfilter (AF) setzt zwar ganz woanders an, er lässt sich aber wie Kantenglättung bei älteren Titeln selbst auf Mittelklassegrafikkarten problemlos hinzuschalten. Besonders in Rennspielen, in denen sich die Muster der Rennstrecke über eine längere Strecke ähneln, verlieren weit entfernte Texturen üblicherweise erkennbar an Schärfe. Die anisotrope Filterung sorgt für eine annähernd gleiche Darstellung aller Texturen, egal, wie weit sie entfernt sind – der optische Vorteil ist enorm. War früher die anisotrope Filterung durch die fehlende Speicherbandbreite der Grafikkarten ein wahrer Leistungsfresser, verlieren moderne Karten nur noch vergleichsweise wenig Leistung durch AF. HW



Ohne Kantenglättung haben Schrägen **hässliche Kanten** (rechts).



Scharfe Texturen dank anisotroper Filterung (links).

Das physikalisch korrekte Ableben von menschlichen Spielfiguren (»Ragdoll Physiker«) störte schon mehrfach die USK.