

Akte Digital_Anvil/3

Loose Cannon

GameStar sprach mit dem Chef-Programmierer des ehrgeizigen Actionspiels über Tricks bei 3D-Engines – und was sein Projekt mit Black & White zu tun hat.



Jörg Neumann, Chef-Programmierer.

Gewaltige Städte mit fließendem Verkehr auf engen Straßen und breiten Highways. Ein Hinterland aus Bergen und Wäldern. Dutzende von steuerbaren Fahrzeugen mit einem ausgeklügelten Fahrverhalten. Und die Möglichkeit, als moderner Kopfgeldjäger entweder einer Kampagne zu folgen oder auf eigene Faust für Ordnung zu sorgen. Mit **Loose Cannon**, der Mischung aus Action-Rennspiel und Ego-Shooter, hat sich Digital Anvil viel vorgenommen. Für GameStar verrät der aus Heidelberg stammende Chef-Programmierer Jörg Neumann, wie eine glaubwürdige Welt im PC entsteht.

1. Grafik

»Die größte Herausforderung von **Loose Cannon** sind die Ausmaße der Städte«, erzählt Neumann. Jede besteht aus ein paar Millionen Polygonen, entsprechend vielen

»Die größte Herausforderung sind die Ausmaße der Städte.«

Texturen – ohne raffinierte Programmiertricks würde das jeden Highend-Rechner überfordern. »Der effektivste Weg, die Polygon-Anzahl zu senken, ist unser Level-of-Detail-

System.« Damit reduziert sich die Anzahl der Details abhängig von der Entfernung eines Objekts – den Wolkenkratzer am

anderen Ende der Stadt muss man schließlich nicht mehr mit jedem Fenstersims darstellen. Beim Spiel von Digital Anvil schrumpelt etwa ein Jeep in sechs Stufen vom detaillierten Auto in der Nahansicht bis zum vierschötigen Klotz, der in weiter Ferne eine Kreuzung überquert. Das ist kein neuer Trick – aber in **Loose Cannon** merkt es das Auge dank der vielen Zwischenstufen nicht mehr. Zum Vergleich: Die **Quake 3**-Engine kennt nur



Nicht nur Stadt und breiter Highway: Die Engine muss flexibel genug sein, auch **ländliche Gebiete** darstellen zu können.

Echte Welt als Vorbild



Wie bei diesem Turm der University of Texas in Austin (oben) dient im gesamten Bild die echte Welt als Vorbild fürs Spiel (unten).

einen dreistufigen Level-of-Detail für die Darstellung der Figuren.

Ein in vielen Spielen beliebter Trick, allzu große Sichtweiten wie in **Midtown Madness 2** durch sehr enge Straßen und hohe Gebäude zu begrenzen, funktioniert in **Loose Cannon** nur eingeschränkt. Denn die Metropolen sollen wie ihre echten Vorbilder aussehen. Doch manchmal ist das schwierig zu erreichen. Etwa in Austin, der Heimat von Digital Anvil. »Bei uns in der Nähe ist ein See, von dem aus man eigentlich den Sitz des texanischen Gouverneurs erkennt«, erklärt Neumann. »Schade, aber im Spiel war die Entfernung da einfach zu weit«. Jetzt schränken ausnahmsweise ein paar Bäume die allzu große Fernsicht ein.

2. Lichteffekte

»Spiele wie **Unreal Tournament** oder **Quake 3** benutzen Light-Maps, also eine zusätzliche Textur-Schicht mit Lichtinformationen«, sagt Jörg Neumann. Bei **Loose Cannon** würde das aufgrund der großen Spielwelt nicht funktionieren – die Zusatz-Licht-Textur müsste gigantisch groß sein und würde den Hauptspeicher komplett verstopfen. »Wir haben da lange mit herumexperimentiert, sind aber zu keiner brauchba-

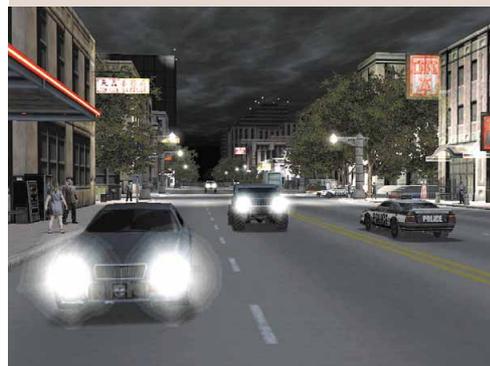
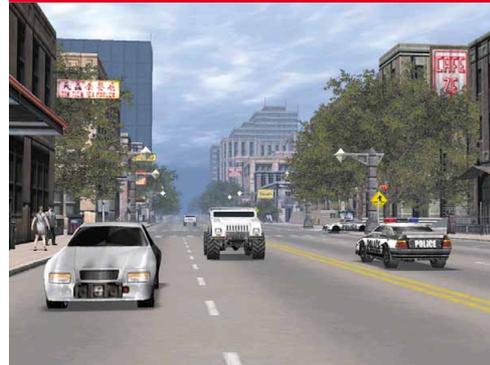
ren Lösung gekommen«, so Neumann. »Bis wir auf den holländischen Spezialisten aufmerksam wurden, der das Problem schon bei **Black & White** gelöst hat«. Der Mann wurde nach Austin eingeflogen und hat ein sogenanntes Light-Mash-System programmiert. Das baut den Ausschnitt, den der Spieler gerade sieht, blitzschnell aus einfachen Polygonen noch mal nach und legt sie über die 3D-Welt. Auf diesen Polygonen befinden sich keine Texturen. Stattdessen enthält jedes Informationen über den auszustrahlenden Lichtwert – und der lässt sich beispielsweise bei Explosionen oder beim Tag-Nachtwechseln vergleichsweise einfach und schnell verändern.

3. Physik

Die Entwickler von **Loose Cannon** wollen eine möglichst glaubwürdige Welt erschaffen. Keine einfache Aufgabe – etwa beim Fahrverhalten: »Das ist manchmal irre, mit was für komplexen Algorithmen man sich rumschlägt. Welche kinetische Energie wird freigesetzt, wenn ein Wagen mit 100 Sachen in einen anderen rauscht, der in ein weiteres Auto, und so weiter?« Das jeweilige Verhalten der Vehikel bestimmen rund 60 Daten – wie Gewicht, Reifenabstand, Federung oder Motorleistung. Aus denen wird errechnet, wie Autos um Kurven schleudern oder bei Hubbeln kurz in der Luft hängen. Das alles sind extrem komplizierte physikalische Berechnungen. Um wenigstens ein bisschen Prozessor-Power zu sparen, verfügen nicht alle Vehikel über das System: das Auto des Spielers zu 100 Prozent, die Kisten von Verbrechern zu etwa 80 Prozent. Unbeteiligte Zivilisten-Autos nutzen die re-

chenintensiven Physiken gar nicht, sondern werden mit einem Standard-Schleuder- und -Rumpelverhalten bewegt, wie es etwa **Driver** generell verwendet. Allerdings – sobald sie in Kontakt mit dem Auto des Spielers kommen, überträgt das Programm blitzschnell das aufwändige PhysiksysteM der wichtigeren Vehikel auf sie. **PS**

Herausforderung Lichteffekte



Tag und Nacht: Ein Light-Mash enthält alle Lichtinformationen; die eigentlichen Texturen bleiben unverändert.



Autos basieren auf einem ausgeklügelten physikalischen Modell, das sich aus rund 60 Einzeldaten zusammensetzt – Offroad-Fahrten werden so zur Herausforderung.