

2013

ISSN 1433-2620 > B 43362 >> 17. Jahrgang >>> www.digitalproduction.com

Published by ATEC

Deutschland € 14,95

Österreich € 17,-

Schweiz sfr 23,-

1

DIGITAL
PRODUCTION

DIGITAL PRODUCTION

MAGAZIN FÜR DIGITALE MEDIENPRODUCTION

JANUAR | FEBRUAR 01|13



Der Hobbit

Jackson jagt die Menagerie durch Neuseeland

Scratch 7

Color Grading +
von Assimilate

MoCap in C4D

Motion und Performance
Capture für jedermann





Teil 2

SpeedTree in der Spieleproduktion

POWERED BY



In der letzten Ausgabe der DP haben wir die allgemeine Funktionalität von SpeedTree erläutert und zudem angefangen einen Baum zu modellieren. In dem zweiten Teil des Workshops gehen wir nun auf die Erstellung der Blätter ein und wollen die Astform noch etwas detailreicher gestalten. Zudem gibt es noch einen Einblick in die verschiedenen zusätzlichen Systeme von SpeedTree, wie zum Beispiel das „Level of Detail“-System, die Wind-Einstellungen oder die Kollisionsgenerierung.

von Sascha Henrichs

Hierfür wollen wir verschiedene Vegetations-Assets produzieren. Diese müssen performant dargestellt werden, also auch ein „Level of Detail“-System unterstützen, und ressourcenschonende Dateigrößen und -anzahlen haben. Zusätz-

lich soll sich die Vegetation noch im Wind wiegen. Nachdem wir also unser Grundgerüst in Form eines Baumstammes und den Ästen erstellt haben, fehlt uns ein entscheidender Bestandteil eines Baumes, nämlich die Baumkrone mit den Blättern.

Blattwerk

Wie schon erwähnt werden Blätter in SpeedTree mit speziellen Nodes erstellt. Fügt man mit der rechten Maustaste auf unseren Level-2-Branch-Node nun einen „StandardRT“-Leaf Node ein, dann sieht der Node-Bereich folgendermaßen aus: An den Zahlen, die zwischen den einzelnen Nodes dargestellt sind, erkennt man wie viele der einzelnen Meshelemente an dem jeweiligen Root erzeugt wurden. So kann man zwischen dem Tree- und dem Trunknode erkennen, dass nur ein Mesh-



Ein einfaches Nodelayout

element als Baumstamm erzeugt wurde. Diese Zahlen sind also keine Polygoncounts, sondern nur Indikatoren für die Anzahl der erzeugten Meshinstanzen oder Elemente. Und wie man sieht, wurden sehr viele „Blattinstanzen“ erzeugt.

Geometrie und Meshimporte

Die Blätter, werden zunächst als sogenannte „Cards“ – besser bekannt als „Billboards“ – dargestellt. Also letztlich einseitige Polygonquads, die sich der Kamera nach ausrichten. Der große Vorteil dieser Darstellungstechnik, ist, dass man mit relativ wenig Polygonaufwand ein sehr dichtes Blattwerk rendern kann.

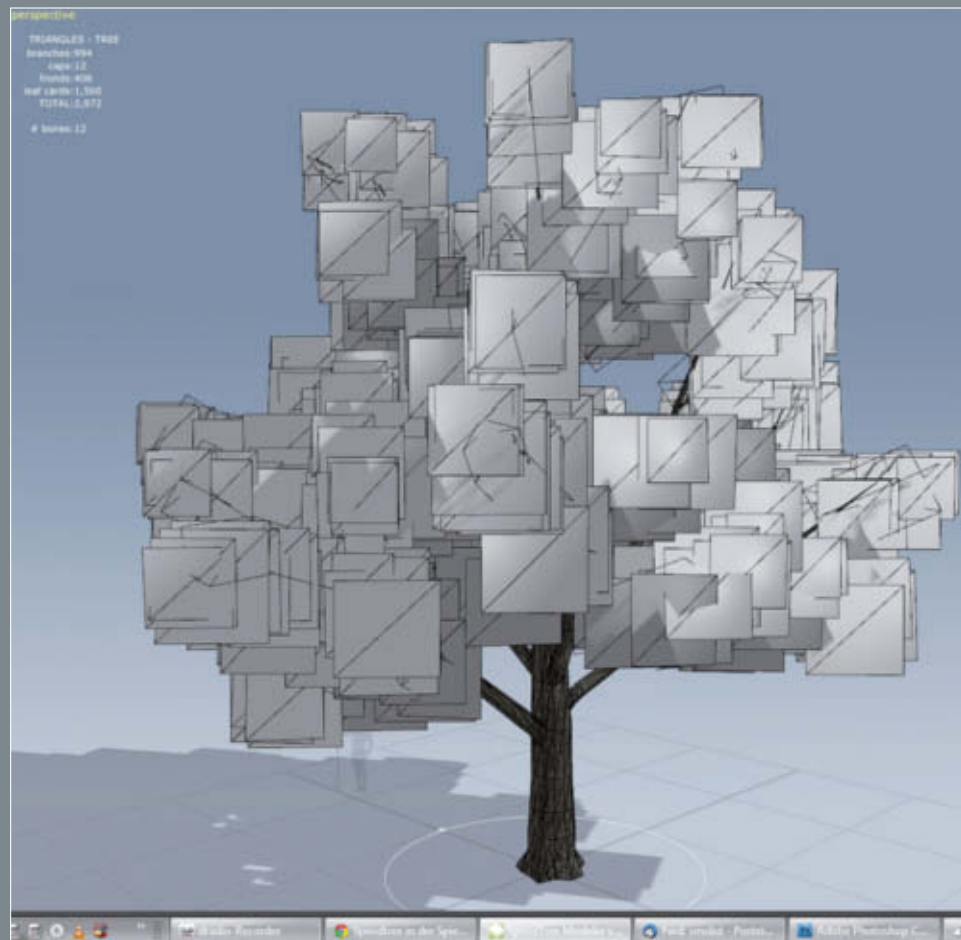
Der große Nachteil ist aber eine hohe Fill-rate, die durch die Überlagerung von etlichen Texturalpha-Bereichen zustandekommt. Für Echtzeitdarstellungen ist dies ein kritischer Punkt. Etliche Pixel müssen hier wieder und wieder auf Durchsichtigkeit getestet werden. Ein weiterer Nachteil ist, dass man es oft gut erkennt, wenn sich die Blätter zur Kamera ausrichten.

Insbesondere wird dies zum Problem, wenn man unter einem Baum herläuft und sich die Planes in einer ruckartigen Bewegung einmal um 180 Grad drehen, um wieder korrekt zur Kamera ausgerichtet zu sein, ohne sich zu überschlagen. Um diese Effekte zu verhindern, sind wir in unseren Spielen dazu übergegangen, komplexere Meshgeometrie für unsere Blattmeshes zu verwenden, die sich nicht zur Kamera ausrichtet.

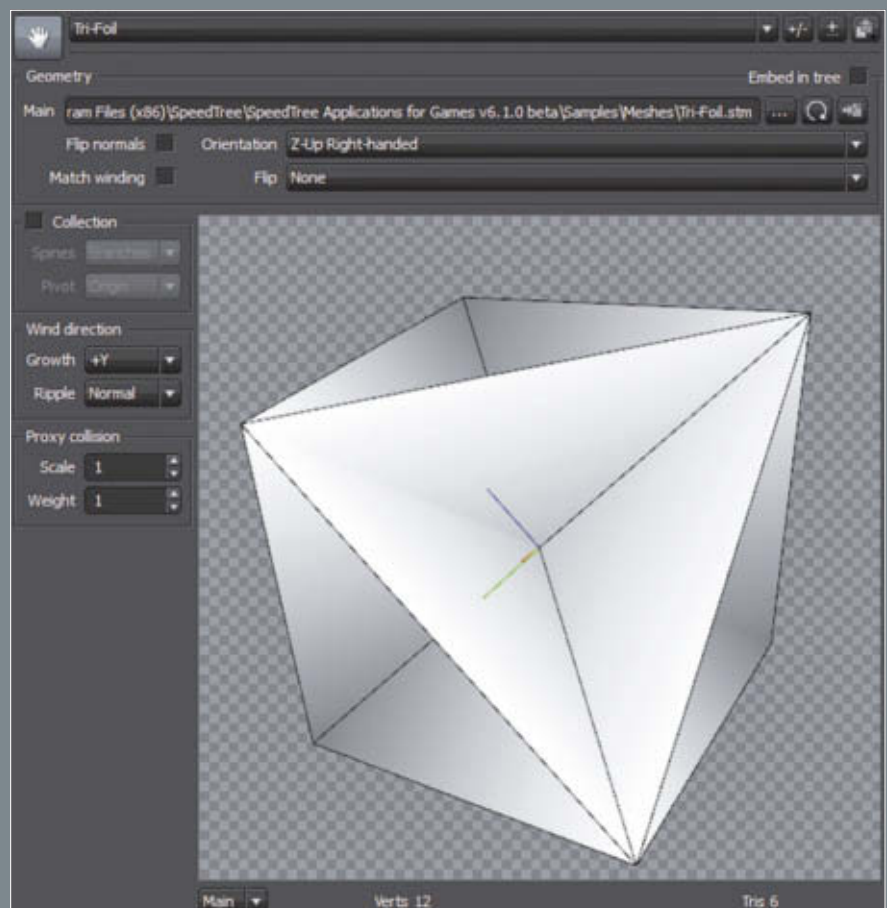
SpeedTree selbst wird mit einigen Beispielen für solche Meshes ausgeliefert. Diese reichen von einfachen Polygonkreuzen bis hin zu clever arrangierten Polygonkonstrukten. Um ein solches Mesh einzuladen, klicken wir im Meshbereich des Assetbrowsers auf das kleine Pluszeichen und navigieren zum Samples/Meshes-Ordner der SpeedTree Root. An dieser Stelle können verschiedene Formate eingeladen werden, auch FBX- oder OBJ-Formate. UVMs, die man für eigens erstellte Meshes angelegt und mit exportiert hat, werden ebenfalls ausgelesen und benutzt.

SpeedTree hat ein eigenes „STM“-Format, das man mit den mitgelieferten 3ds-Max-Scripten selbst exportieren kann. Dieses Format kann veränderte Polygonnormalen exportieren, wie sie für bestimmte Beleuchtungsarten für Blätter im Niedrigpolygonbereich von Vorteil sind. Für Maya gibt es diese Scripte nicht. Wählen wir also das Mesh Tri-Foil.stm und weisen es in den Leaves Properties zu.

Danach wählen wir eine passende Blatt-Textur aus und weisen diese ebenfalls zu. Das Ergebnis lässt allerdings nach wie vor zu wünschen übrig. Die Blätter sind zu einheit-



Unser Baum mit sogenannten „Leaf Cards“. Einfache Polygonflächen, die sich zur Kamera ausrichten.



Eine Vorschau für die eingeladenen Geometrien



Dieser Baum ist sehr unspektakulär und zeigt unrealistischen Blattwuchs.

lich verteilt, der Stamm ist unnatürlich gerade und viele der Level 1 Branches haben vielleicht noch nicht die gewünschte Ausrichtung, was man beim Rotieren des Baumes schnell erkennt. Doch genau hier wird es letztlich kreativ und spannend.

Von Formen ...

SpeedTree bietet eine Menge Parameter mit denen man die Form, Ausrichtung und Anzahl der Äste und Blätter beeinflussen kann. Im Folgenden werde ich die Namen der Para-

meter und deren Funktion erwähnen, die ich für unser Beispiel verändere. Die Notation der tatsächlichen Zahlenwerte wäre an der Stelle nur verwirrend.

Dem SpeedTree-Neuling sei nahegelegt, unbedingt selbst an den Werten zu drehen und zu experimentieren, um ein Gefühl für die Parameter zu bekommen. Hat man erstmal das System verinnerlicht und für jeden Parameter ein Bild von seiner Funktion vor Augen, ist es ein Leichtes zu erkennen, warum sich die Geometrie so verhält, wie sie es tut.

Wenden wir uns also wieder unserem Baum zu, fällt auf, dass dieser organischer wirken und interessantere Silhouetten werfen sollte. Zudem möchte die Baumkrone voller und natürlicher wirken. Das alles unter dem Gesichtspunkt, Polygoneffizienz zu wahren. Zunächst wollen wir aber unsere Blätter ausblenden und uns nur den Ästen zuwenden.

Schauen wir uns im Netz um, finden wir eine Menge Beispiele an spannenden Baum-silhouetten. Analysieren wir diese, fällt uns auf, dass wir unseren Baumstamm zunächst unregelmäßiger entwerfen müssen, um das Gesamtshape interessanter zu gestalten.

Unten sehen wir nun unseren Baum, nachdem wir ihn mit den prozeduralen „Modellierungswerkzeugen“ bearbeitet haben. Im Folgenden gehe ich auf die wichtigsten Schritte ein, die dafür nötig waren.

Der Wert „Disturbance“ den wir zunächst ändern, findet sich im „Spine“ Rollout der Baumstamm-Properties. Dadurch wird der Baumstamm an einigen Punkten in zufällige



Vergleich von unserer Astgeometrie mit einer interessanten natürlichen Form

Derselbe Baumstamm in SpeedTree, nach unseren Verbesserungen



Richtungen gebogen. Haben wir dies getan, möchten wir in den Stamm noch eine Astgabelung integrieren. Die „Bifurcation“ ist ein eigenes Rollout. Stellen wir hier die Bifurcation „Chance“ auf den Wert 1, gehen wir sicher, dass eine Gabelung in jedem Fall stattfindet.

Das „Spot“-Property kann die Gabelung entlang der Stammlänge verschieben. Um diesen zweiten Ast, der durch die Astgabelung entstand, nun auf einer gewissen Höhe abzubrechen, stellen wir das Property „Break Chance“ im „Spine“-Rollout auf 1. Im Bifurcation-Bereich geben wir an, welcher Ast an welcher Stelle abgebrochen werden soll. Alle Stellen, an denen ein Ast nicht nur zu einem Vertex am Endpunkt zusammenläuft, nennt man „Caps“. Diese können zum einen durch Astbruch entstehen oder durch eine Astdicke, die am Astende noch nicht auf 0 zurückgefahren wurde. Wie es zum Beispiel an dem anderen Ende unseres Hauptastes der Fall ist.

Um nun an diesen Caps schöne abgesplitterte Abbruchstellen zu bekommen, brauchen wir noch ein paar mehr Polygonsegmente in den „Cap“-Bereichen, die wir im „Segments“-Rollout und dort unter „Cap“ einstellen können. Ein entsprechendes „Cap“-Material erstellen wir uns auf dieselbe Weise wie die anderen, und weisen dieses den „Cap“-Bereichen zu.

Die Wurzelansätze, die man im unteren Bereich des Stammes erkennen kann, werden in SpeedTree „Flares“ genannt. Diese finden wir im „Branch“-Rollout unter eben jener Bezeichnung. Haben wir all diese Veränderungen vorgenommen, sieht unser Baumstamm schon interessanter aus. Dennoch fehlt den „Level 1“-Branches noch dieselbe Behandlung. Zudem wollen wir auch die Anzahl unserer generierten Äste verringern und tragen im „Generation“-Rollout einen geringeren Wert ein. Hat man dies alles erledigt, macht es bei den Ästen noch Sinn, diese per Hand an passende Stellen zu schieben.

Man hat in SpeedTree jederzeit die Möglichkeit, nicht nur einen Node in seiner Gesamtheit zu bearbeiten, sondern man hat Zugriff auf jede von ihm erzeugte Meshinstanz, indem man „Allow Node Editing“ anschaltet. Dies erreichen wir mit dem Shortcut TAB oder mit einem kleinen Icon oberhalb des Viewport-Bereiches. Ist dieses aktiviert, können wir einzelne Äste im Viewport anwählen, drehen, verschieben oder auch in der Größe verändern.

Man hat nun auch die Möglichkeit, auf jede prozedurale Eigenschaft einer Meshinstanz einen Offset-Wert entweder hinzu- oder abzuziehen. Man kann also einzelne Äste gezielt in ihren Einstellungen zum Node abweichen lassen. So ist es zum Beispiel möglich einen Ast ganz besonders dünn im Gegensatz zu den anderen erscheinen zu lassen.

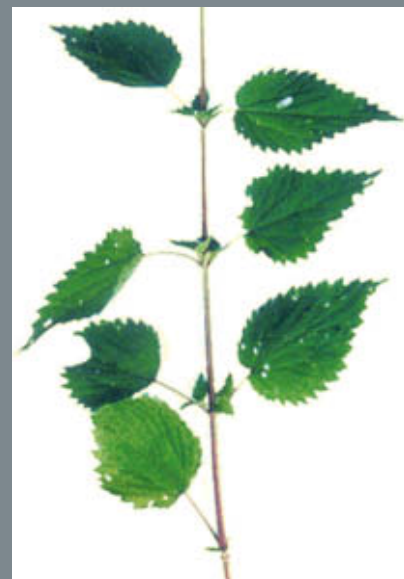
... Blättern ...

Hat man nun die Äste und Fronds entsprechend arrangiert und die Blätter wieder sichtbar geschaltet, so sehen wir eine unnatürliche Verteilung der Blätter und eventuell auch eine suboptimale Silhouette. Es ist also notwendig, auch hier noch einige Veränderungen vorzunehmen. Unten sehen wir nun den Vergleich von der unveränderten Version zu einer bearbeiteten Version. Im Folgenden gehen wir kurz auf die vorgenommenen Änderungen ein.

Einige Äste wurden gedreht und skaliert. Zudem gibt es auch noch einen zusätzlichen Frond Node, der an den Baumstamm angefügt wurde. Dieser erzeugt noch weitere drei Fronds mit Zweigtexturen an dem Baumstamm, denen wir jedoch keinen Blattwuchs mitgeben. Das lockert das Gesamtbild etwas auf, indem die sonst verdeckten Zweige hier in den Vordergrund treten. Die Blattinstanzen selbst wurden in ihrer Anzahl reduziert. Dafür haben wir zunächst den „Style“ in den „Generation“-Properties von „Relative“ auf „Absolut“ gestellt. Dadurch ist gewährleistet, dass wir eine exakte Anzahl von generierten Blättern in der „Frequency“ angeben können, welche sich ebenfalls in den „Generation“-Properties findet. Steht der „Style“ auf relative, so würde das Programm anhand der eingestellten Frequenz und der Astlänge, an der die Blätter erstellt werden, selbst berechnen wie viele Blätter erzeugt werden sollen.

Zudem schalten wir „Orientation“ und „Steps“ aus. Mit der „Orientation“ können wir Blätter in immer gleichen Abständen zueinander platzieren und in den Steps kann man sagen, dass an den Erstellungspunkten gleich mehrere Instanzen nebeneinander erstellt werden sollen. Dies ist beispielsweise für Stengelpflanzen jeglicher Art von Bedeutung, die an ihren Trieben jeweils zwei oder mehr Blätter an derselben Stelle austreiben. Für unseren Baum ist dies allerdings kontraproduktiv, da die Meshinstanzen so nah beieinander liegen, dass diese unnötige Polygone erzeugen. Für die letzte Veränderung an den Blättern, widmen wir uns den „Placement“-Properties.

Einige Blätter, oder besser „Leafmeshes“ haben nach unserer Reduzierung der Blattdichte keine Verbindung mehr zu den Fronds, stehen also frei im Raum. Mithilfe der Werte „Maximum“ und „Minimum“ in den „Placement“-Properties können wir die Ent-



Blattanordnung an Stengelpflanzen

fernung der Blattmeshes zu den Zweigen angeben. Stehen danach immer noch Blattmeshes im Raum, können wir diese explizit anwählen, verschieben oder löschen.

Haben wir dies getan, sind alle relevanten Änderungen vorgenommen und unser Baum ist fertig modelliert. Es gibt allerdings noch fortgeschrittenere Möglichkeiten, Formen in SpeedTree zu beeinflussen. Auch diese wollen wir kurz besprochen werden.

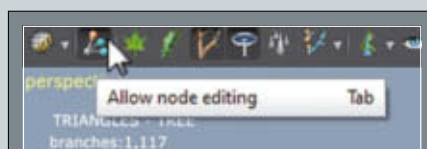
... und Forces

Eine sehr interessante Möglichkeit, Geometrien in SpeedTree zu verformen, bieten die sogenannten „Forces“, also Kräfte, die man in die Szene einfügen kann und die entsprechend ihrer Art den Baum formen können.

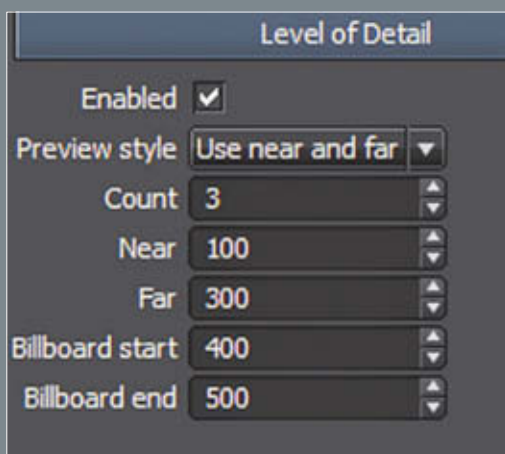
Es gibt zum Beispiel die „Magnet“-Force, die eine vorhandene Geometrie zu ihrem Mittelpunkt ziehen kann. Es gibt „Directions“, die den ganzen Baum oder einzelne Äste wie bei einer Trauerweide direktional wegdrücken können. Dann gibt es auch eine „Twist“-Force, die den Ast in Spiralen wachsen lassen kann. Neben weiteren solcher prozeduralen Forces gibt es aber auch noch die „Mesh“-Force. Hier kann eine eigene eingeladene Geometrie als Kraft wirken. Man kann diese Geometrie beliebig in der Szene platzieren und skalieren.

Wählt man sie an, kann man in ihren Eigenschaften angeben, wie dieses Mesh auf unseren Baum wirken soll. Eine sehr interessante Verwendung finden solche Mesh-Forces in der Benutzung als Kollisionsgeometrie für unseren Baum. In diesem Fall würden die Äste von der Oberfläche unserer Force angezogen, könnten diese aber gleichzeitig auch nicht durchdringen.

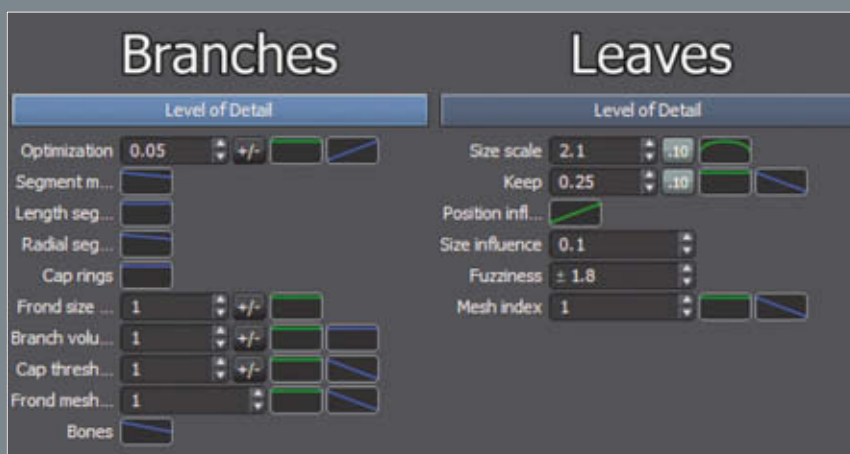
Gibt man diesen Mesh-Forces dann auch noch ein Material, so werden diese auch im SpeedTree Modeller korrekt dargestellt.



Im Node-Edit-Modus kann man die einzelnen Äste oder Blätter verändern.



Die LOD-Stufen-Einstellung im Treenode



Die LOD Properties von Branch- und Leavenodes



Schlecht eingestelltes LOD

Ein Baum wie gemalt

In einigen Fällen reichen all die prozeduralen Modellierungsmöglichkeiten aber nicht aus. Dann möchte man vielleicht auf die komfortable Möglichkeit zurückgreifen, Äste mit der Maus oder dem Wacom Tablet zu malen. Auch dies ist in SpeedTree möglich.

Wir müssen nur die Space-Taste drücken und mit linker Maustaste einen Ast im Screenspace aufziehen. Ist es der erste Ast eines Baumes, muss man dies in dem Treenode-Radius tun, also in dem kleinen Kreis am Boden. Ansonsten kann man keinen Ast generieren. Letztlich malen wir nun einen Spline auf dem automatisch eine Ast-Geometrie erstellt wird.

Dadurch, dass nun komplette künstlerische Freiheit gegeben ist, können wir mit diesem Feature auch Bäume er-

stellen, die einem konkreten Konzept folgen. SpeedTree liefert einige Beispiele mit. Etwa die „Beach Palm“.

Es ist nicht alles Gold, was glänzt

So schön sich auch das Arbeiten mit dem SpeedTree Modeler gestaltet, es gibt einige Punkte, die einem das Leben schwer machen können.

Die Kollision

In der Regel sollte eine Spielfigur mit ihrer Umgebung kollidieren. SpeedTree ist auch in der Lage, die Bäume mit speziellen Kollisionsobjekten auszustatten. Zur Auswahl stehen „Sphären“ und „Kapseln“.

Diese können in der verwendeten Spielengine durch einfache Radius und Positionsberechnungen vom Physiksystem abgeleitet werden. Es sind also keine Ressourcen- und performancebelastenden Physik Meshes notwendig. Der in SpeedTree integrierte Collision Generator bietet die Möglichkeit, diese Kapseln und Sphären automatisch am Baum zu generieren. Dafür suchen wir zunächst das Icon im oberen Viewport-Bereich, welches drei rosa Bälle beinhaltet. Dort klicken wir auf den „Rollout“-Pfeil und führen den Befehl „Generate Collision Primitives“ aus. Nach einigen Detail-Einstellungen, die man hier vornehmen kann, liefert dieser Mechanismus allerdings ungenügende Ergebnisse.

Wir sehen, dass die Kapsel vom Baumstamm viel zu groß ist. Zudem sind etliche kleinere Kapseln erzeugt worden, die sich fast gänzlich überlagern. Es lässt sich zusammenfassen, dass die automatische Erzeugung der Kapseln in der Regel nicht zu gebrauchen ist. Und man kann diese Kapseln zwar von Hand setzen, aber letztlich vermisst man hier „Snapping“-Mechanismen oder ein zügig zu bedienendes Movement-System. Denn das derzeitige System, sieht zwar Keyboard Shortcuts für das Movement, die Skalierung und die Rotation vor, allerdings sind diese gemäß des englischen Textur Layouts auf Z,X,C gemappt. Was zur Folge hat, dass man sich beim Drücken der Z-Taste regelmäßig vertippt, da diese im deutschen Layout in der Mitte des Keyboards sitzt. Bisher habe ich auch keine Möglichkeit gefunden, das Tastatur-Layout zu verändern. So wird das Erzeugen von Kollisionen für SpeedTrees zur Geduldsprobe. Hat man in seinem Spiel einen Fundus von 100 oder mehr Bäumen,

In Computerspielen setzt man diese Möglichkeit nicht allzu oft ein, da man diese speziell verformten Bäume oft nur einmal an einer ganz bestimmten Stelle plazieren kann. Zum Beispiel wenn das Wurzelwerk eines Baumes um eine einzigartige Architektur herum wächst.

Dennoch lohnt es sich unbedingt für ausgewählte Spielbereiche, Arbeit und Ressourcen zu investieren, um Erinnerungswürdige Bilder zu erschaffen.

sollte man so diszipliniert sein und direkt bei Erstellung eines jeden Baumes die Kollision mitliefern. Diese nachträglich hinzuzufügen ist eine zermürbende Aufgabe.

Die LOD-Stufen

Ähnlich verhält es sich mit den LOD-Stufen. Diese zu erstellen ist oftmals aufwendig, da auch diese Einstellung zunächst prozedural, also basierend auf Zahlenwerten, Wahrscheinlichkeiten und Kurven erfolgt.

Möchte man das LOD'ing im SpeedTree Modeller sehen und einstellen, so wählt man zunächst den Treenode aus und stellt diesen im LOD Preview Style auf „Use near and far“. Dann kann man sich Live im Viewport anschauen, wie der LOD'ing-Mechanismus auf unseren Baum wirkt. Zoomt man aus dem Bild heraus, wird auf Basis der Entfernungswerte das Erscheinungsbild und die Auflösung unseres Baumes geändert. Ist man hier angekommen, kann man nun jeden einzelnen Node des Baumes einstellen. Dabei bietet das System einige Tücken. Gerade bei Computerspielen ist man darauf angewiesen, niedrigere LOD-Stufen schon bei geringen Entfernungen zur Spielerkamera einzublenden. Allerdings sind die Templates und die Bäume der Library oftmals so eingestellt, dass signifikante Teile der Fronds und Blätter bei Entfernung kleiner skaliert werden, bis sie verschwinden, und einige wenige Blätter davon, größer skaliert werden, um die entstehenden Lücken wieder zu füllen. Das hat unter Umständen bei geringen Kameradistanzen den Effekt, dass einige Blattgruppen scheinbar wachsen, wenn man sich auf sie zu bewegt.

Diese Effekte kann man zwar mit entsprechender Einstellung unterbinden. Aber da man seinen Baum fast immer aus fertigen Bäumen oder SpeedTree Templates zusammenbaut, muss man dies auch tatsächlich für jeden Baum neu einstellen. Hauptsächlich sind es die Werte „Branch Volume“ und „Fronde Size“ bei Ästen, bei den Zweigen der Wert „Size Scale“ und deren Kurven. Es bietet sich also letztlich an, eigene Templates zu erstellen, die auf die speziellen Bedürfnisse angepasst sind.

Die Einheiten

Die SpeedTree-Bäume der mitgelieferten Library sind alle grob in „Feet“-Einheiten modelliert. Ansonsten bietet das Programm keine verständlich angegebenen Einheiten an. In 3ds Max können wir einfach das gesamte Programm auf die gewünschten Einheiten umstellen. Dies fehlt in SpeedTree. Teils, weil man es oft mit generischen Einheiten zu tun hat, teils, weil das gesamte interne Konzept von SpeedTree nun mal auf „Feet“ und „Inches“ basiert. Stattdessen gibt es eine An-

zahl von Tools, die es dem Benutzer ermöglichen, Einheiten zu konvertieren. Angefangen von Scripten für 3ds Max, über FBX-Konversionen im Export-Dialog bis hin zu internen SpeedTree Modelertools, die Einheiten konvertieren können.

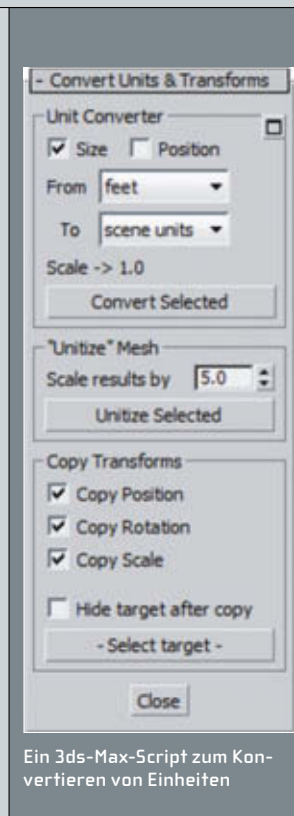
Wer einmal mit der Konvertierung von Einheiten zu tun gehabt hat, wird wissen, dass dies schnell zu allgemeiner Verwirrung führt. Man muss einfach viel Gedankenarbeit leisten, um einen Überblick darüber zu behalten, was man überhaupt tut. Dies ist leider alles andere als eingängig und intuitiv. Und man wünscht sich sehr schnell eine Einstellung in den Präferenzen, die das gesamte Programm auf die gewünschte Einheit umstellt. Insbesondere wird es schwierig, wenn man Kollisionsgeometrie (für eine Meshforce) in den Modeller einladen möchte. Dabei ist es natürlich gewünscht, dass die tatsächlichen Ausmaße des Objektes gewahrt bleiben.

Stattdessen passiert es aber ständig, dass Meshes zu groß oder viel zu klein im Modeller ankommen. Je nachdem, welches Format und welche Einstellungen man beim Export der Geometrie verwendet hat. Ich selbst habe nach einigem Ausprobieren herausgefunden, dass in 3ds Max exportierte Meshes im FBX-Format korrekt exportiert werden können, wenn man im FBX „Save“-Dialog, „scene units converted to: -> Inches“ einstellt.

Der Wind

Lange Zeit war der Wind in SpeedTree tatsächlich ein Sorgenkind. Und eigentlich sei er nur an dieser Stelle erwähnt, weil die Neuerung, die die Windeinstellung für einen Baum vereinfacht, wirklich erst seit kurzer Zeit in das Programm implementiert ist.

Ähnlich wie bei den LOD-Stufen musste man jeden Node einzeln und von Hand einstellen, wie er auf einen Wind zu reagieren hat. Zusätzlich gab es dann eine Winddatei, in der die Daten für diesen Wind verankert waren. All die Einstellungsmöglichkeiten aufzuzählen,



Ein 3ds-Max-Script zum Konvertieren von Einheiten

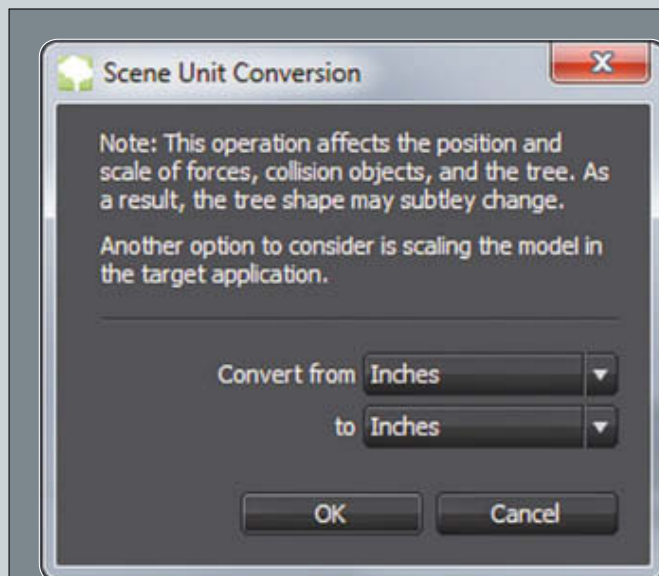
würde hier den Rahmen sprengen. Es sei an der Stelle nur erwähnt, dass es seit kurzer Zeit einen sehr intuitiv zu bedienenden „Wind Wizard“ gibt. Jeder Baum in SpeedTree 6.2 hat nun eigene Windparameter und reagiert nicht mehr auf eine global wirkende Winddatei. Diese Parameter werden in einigen wenigen Mausklicks mit dem Windwizard sehr brauchbar eingestellt und man muss nunmehr nur noch Kleinigkeiten an den Windeinstellungen verändern. Man gibt an, um welche Art von Baum es sich handelt, ob Palme, Laubbaum oder Busch, und welche Art von Blättern vorliegt. Kleine Blätter, so wie die der Birke, oder große Blätter, wie etwa die des Ahorn. Sind die entspre-

chenden Vertexshader, die die Bewegungen am Baum dann abbilden, in die Spieleengine integriert, werden alle Windeinstellungen letztlich auch in der Engine angezeigt.

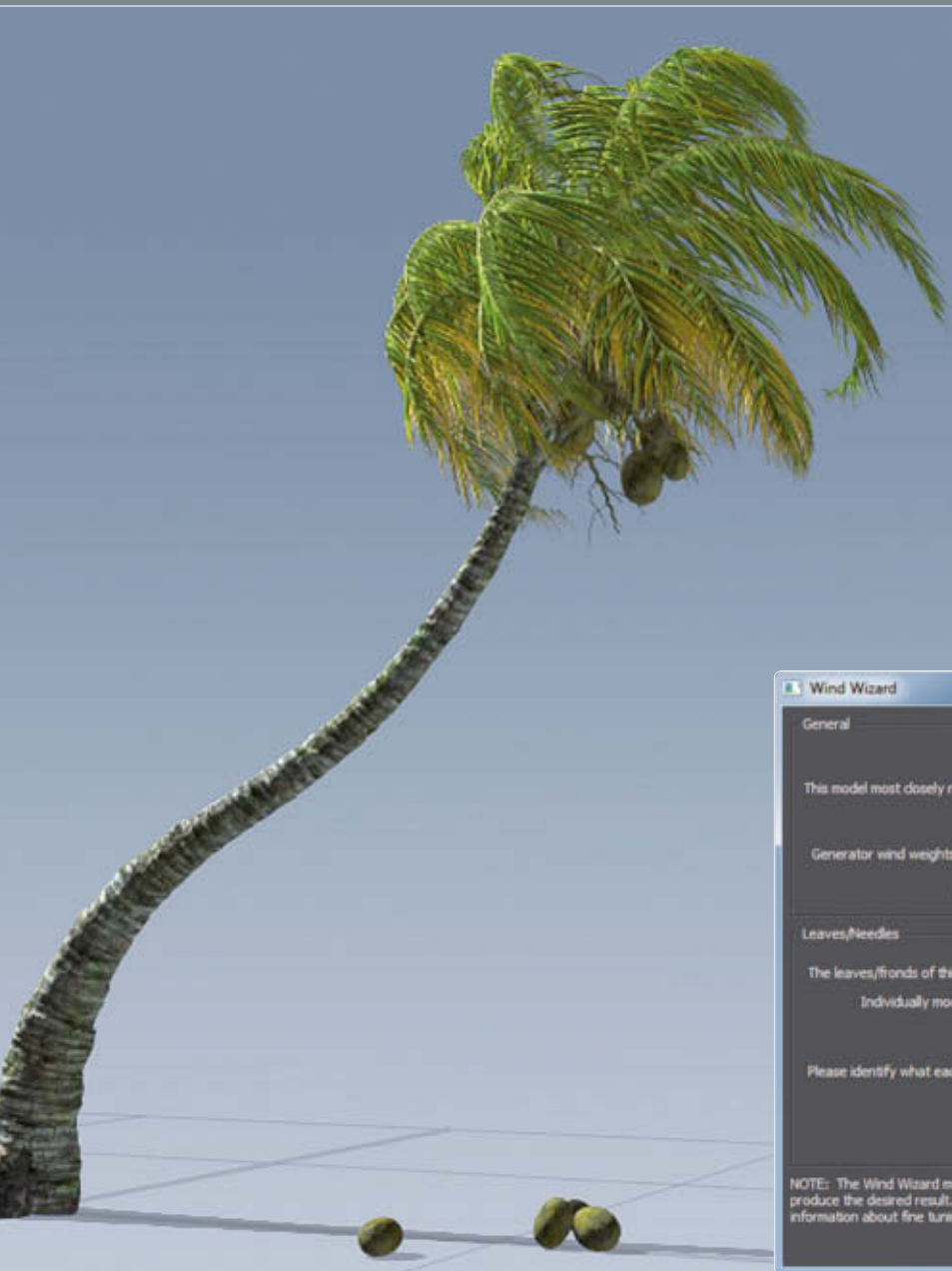
Der Compiler

Ein fertiger SpeedTree-Baum besteht in der Regel aus einer Baum-Datei und einigen Texturen. Die gesamte SpeedTree-Vegetation eines fertigen Spiels umfasst aber oft viele verschiedene Baumgeometrien und etliche Texturen.

Nun ist es für einen Realtime Renderer (eine Spieleengine ist ja nichts anderes) günstiger, einige wenige und dafür große Texturen aufzurufen als hunderte von kleinen



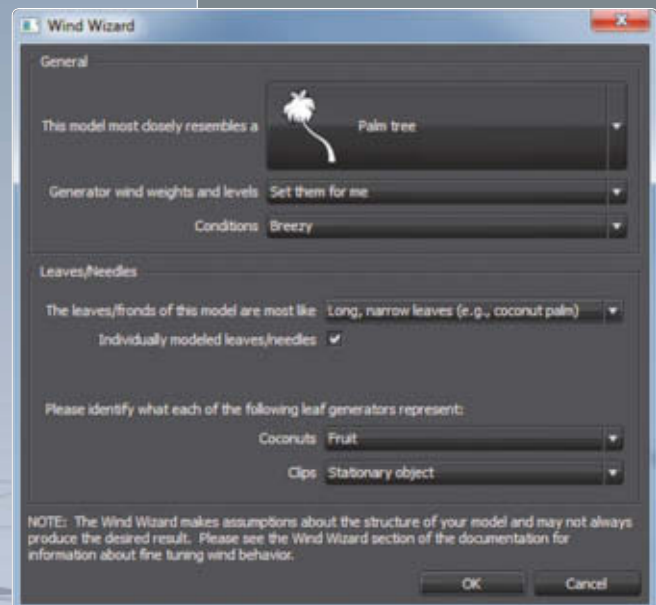
Das SpeedTree-interne Tool zum Konvertieren der Einheiten



Eine Palme bei stürmischem Wetter

sende Bild aus dem Atlas aus. Bei einheitlich geformten Bäumen macht es oft keinen Sinn, mehr als zwei oder drei Ansichten für die Billboards rendern zu lassen. In dem Bildbeispiel sieht man, dass die Fichte von allen Seiten fast gleich aussieht. Bei sehr gebogenen Bäumen aber ist es notwendig, mehrere Ansichten zu haben, denn man kann nie wissen, von welcher Seite sich der Spieler dem Baum nähert. Hat man bei der Palme nur eine nach rechts gebogene Ansicht für das Billboard, aber man nähert sich dem Baum von der anderen Seite, so erscheint der Baum beim Übergang zur komplexen Geometrie plötzlich nach links gebogen. Daher bräuchte ein solcher Baum auch noch eine Billboard-Ansicht mit der nach links gebogenen Palme.

Ein weiteres Feature des Billboard Compilers, ist, dass er direkt beim Erstellen der Bilder entsprechend zugeschnittene Poly-



Der neue „Wind Wizard“ von SpeedTree

Texturen. Und genau dies ist eine der Hauptaufgaben des SpeedTree Compilers.

Hier werden die verschiedenen Texturen mehrerer Bäume auf wenige große Texturatlanten zusammengelegt und die UV-Koordinaten der Bäume im Compileprozess automatisch auf diese neuen Texturpositionen umgemappt. Zusätzlich werden alle prozeduralen Baummeshes in einfache Vertexpositionen umgewandelt, die direkt von der Engine ausgelesen werden können.

Diese kompilierten SpeedTrees sind im Nachhinein nicht mehr vom Modeller einzulesen und auch nicht mehr zu verändern. Sind nachträgliche Änderungen an SpeedTrees gewünscht, muss die Rohdatei verändert und neu gespeichert und der ganze Compileprozess nochmals durchgeführt werden.

Die letzte LOD-Stufe

Doch nicht nur die Meshes werden speichersparend abgespeichert und die Texturen zusammengefasst. Es werden zusätzlich auch alle Billboards der Bäume gerendert und gleichzeitig auf Atlanten verteilt.

Billboards sind nichts anderes als die letzte LOD-Stufe, die von einem Baum in der Engine gerendert wird. Sie besteht in der Regel aus einer Polygonplane, also zwei Triangles, die in der Engine zur Kamera ausgerichtet werden. Optional kann man auch eine horizontale Plane zusätzlich rendern lassen, falls man in einem Spiel auch von oben auf die Bäume herabblicken kann. Man kann einstellen, wie viele Ansichten man pro Baum berechnen lassen möchte. Die Billboards wählen sich dann je nach Kamerawinkel das nächst pas-

gonplanes erzeugt. Denn es kann sein, dass eine reine rechteckige Form für sehr ausladende Baumformen, zu viel „leeren“ Platz auf einem Billboard generiert. Dieser Platz wird mit einer Alpha-Maskierung gefüllt und muss in der Engine stets auf Sichtbarkeit überprüft werden. Im Falle eines Waldes müssten sogar mehrere Flächen hintereinander auf Sichtbarkeit überprüft werden, was letztlich die Füllrate belastet.

Die Füllrate ist die Anzahl der Texel (in der Engine dargestellte Texturpixel), die eine Graffikarte in einer Sekunde berechnen kann. Übersteigt eine Szene diese Anzahl von Texeln dadurch, dass ein und derselbe Bildschirmbereich immer wieder auf Sichtbarkeit getestet werden soll (wie es bei vielen Alpha Planes hintereinander der Fall ist), so bricht die Performance ein. Die Entwickler von SpeedTree

wissen dies, und haben im Billboardgenerator die Möglichkeit gegeben, dass ungenutzte Texturbereiche durch die Billboardgeometrie abgeschnitten werden. Denn da wo keine Alphatextur auf einer Geometrie liegt, muss auch nicht auf Sichtbarkeit getestet werden. Dies wiederum entlastet die Füllrate.

Die Baumbibliothek

Abschließend sei wollen wir noch ein paar Worte über die umfassende Baumbibliothek verlieren. Es werden in dieser Library etliche SpeedTrees in verschiedenen Kategorien angeboten.

Die Hauptkategorien sind

1. Broadleaves (Laubbäume)
2. Conifers (Nadelbäume)
3. Palms_&_Cacti (Palmen und Kakteen)
4. Shrubs_&_Flowers (Büsche und Blumen)

Zudem gibt es noch Spezial-Kategorien wie

5. Marine
6. Hand Drawn
7. Miscellaneous_&_Fantasy

Insgesamt gibt es circa 180 Bäume in der Library. Von den Spezialkategorien abgesehen sind diese Bäume möglichst botanisch korrekt umgesetzt. Zu einem Baum gehören immer alle Texturen, alle zusätzlichen Meshes, wenn welche verwendet wurden, und natürlich die Baumdatei (*.spm) selbst. Zusätzlich kann es sein, dass ein Baum in verschiedenen Polygonauflösungen und in verschiedenen Formen vorliegt. So kann ein Baum in einer Frühlingsvariante, einer Herbst- und einer Wintervariante vorliegen.

Es sei anzumerken, dass man in der Regel keinen der Bäume aus der Bibliothek eins zu eins in seinem Spiel einbauen möchte. Man muss immer noch Polygoncounts tweaken oder LOD-Stufen anpassen, Shapes interessanter gestalten oder Kollisionsobjekte anpassen. Aber als Anfangspunkt kann so ein Baum immer dienen.

Natürlich ist es mehr als hilfreich, die oft sehr guten Texturen mitgeliefert zu bekommen. Die kann man jederzeit auch für eigene Baumkreationen einsetzen und es erspart eine Menge Zeit, diese nicht selbst erstellen zu müssen. Es gibt zwar immer mal eine Textur, die man selbst gemacht hat und die man statt einer mitgelieferten lieber benutzen möchte, aber in der Regel bleibt man aus Zeitmangel bei den Texturen der Baumbibliothek hängen. Vorausgesetzt, dass Artstyle des Spieles lässt dies zu.

Fazit

Seit vielen Jahren arbeite ich nun schon mit SpeedTree. So um das Jahr 2004 war das Pro-

dukt bei weitem noch nicht so ausgereift. Viele Features, Bugfixes sowie ein Facelift sind mit den Jahren hinzugekommen, aber dennoch hat sich an der grundsätzlichen Art der Erstellung der Bäume kaum etwas geändert. Das ist der Grund, warum SpeedTree bis heute so erfolgreich ist. Denn die Geschwindigkeit, mit der dem prozeduralen Modeler gute Ergebnisse abgerufen werden können, ist immens. Ist das Produkt einmal fehlerfrei in die Engine integriert, ist es auch auf Seiten der Programmierung in der Regel vollkommen wartungsfrei. Die Bäume werden durch den Compiling-Prozess optimiert und korrekt eingestellte LODs wirken sich nochmals positiv auf die Spielperformance auf.

Doch Vorsicht! Problematisch kann es werden, wenn man es sich auf Datenseite zu einfach macht. Denn es ist durchaus noch Arbeit, einen Baum komplett zu produzieren. Der Baum selbst, das LOD'ing und die Kollision wollen für jedes Gewächs korrekt eingestellt sein. Groß ist die Versuchung, einen mühseligen und vom Workflow her noch nicht optimal eingestellten Arbeitsschritt etwa die Kollisions- oder LOD-Erstellung vor sich herzuschieben und erst später zu bearbeiten. So kann sich unbemerkt ein Haufen

Arbeit ansammeln, der in keiner Planung auftaucht. Dieses Problem ist allerdings hausgemacht und schmälert den Wert von SpeedTree nicht. Wenn IDV noch ein funktionierendes metrisches Einheitensystem einbaut, dann gibt es nur noch wenige Bereiche, die weiter zu verbessern wären.

Wer also nach einer Lösung für den Vegetationsbedarf in seinem Spiel sucht, kommt heutzutage kaum noch an SpeedTree vorbei. Im Übrigen: SpeedTree ist seit langer Zeit integraler Bestandteil des kostenlosen Unreal Development Kits (UDK) von Epic. Der Modeler und der Compiler werden stets mit installiert und stehen dann in einer nicht ganz aktuellen Version in vollem Umfang zur Verfügung. Wer also SpeedTree for Games in einer Spiele-engine testen will, kann hier den kompletten Workflow kennenlernen. In diesem Sinne viel Spaß beim Pflanzen!

» ei



Ein Mangrovenbaum aus der SpeedTree Library



Sascha Henrichs ist seit 1998 Environment-Grafiker bei Piranha Bytes in Essen und hat an allen PB Games mitgewirkt. Er unterrichtet zudem seit 2007 als freier Dozent hauptsächlich an der Games Academy. Arbeiten und Tutorials sind zu finden unter www.saschahenrichs.blogspot.com.