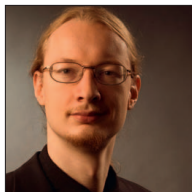


1.000 Schwerter in 2 Tagen

Generierung von Waffen-Modellen

Der Mathematiker und Game-Artist Jochen Winkler erklärt im Workshop Schritt für Schritt, wie Sie mit Algorithmen sowie wenigen handgebaute Klinge und Griffen Tausende unterschiedliche Schwerter generieren.



Jochen Winkler

ist selbstständiger Game Artist.

Nach sechs Semestern Mathematikstudium wechselte Jochen Winkler auf die Games Academy in Berlin und absolvierte die Ausbildung zum Game Artist. Seitdem ist er als Grafiker tätig und nutzt seine Doppelqualifikation, um ungewöhnliche Grafikprobleme mit Algorithmen und mathematischen Formeln zu lösen. Seit Anfang 2011 ist er selbstständig und kümmert sich nach eigener Beschreibung um »Spiele und algorithmische Grafik«. Jochen ist unter info@jochen-winkler.com erreichbar und freut sich über Anfragen – egal ob Sie 3.000 Aliens oder einfach nur etwas Hilfe beim Bau Ihres Tutorials benötigen.

www
makinggames.de

• Cinematics aus Ingame-Assets

Spätestens seit dem Erfolg von Diablo 1 wissen wir: Waffensammeln macht Spaß! Irgendwann ist diese Erkenntnis allerdings mit der Branche durchgegangen. Angehende Game Designer wünschen sich meist mehr Waffen, als ein Team jemals umsetzen könnte. Marketingpartner wünschen sich große Zahlen, um sie auf die Packungsrückseite zu drucken. Und die Geldgeber? Die freuen sich ebenso, denn »Spielspaß« kann man nicht per Vertrag einfordern. »800 Waffen« allerdings schon.

Doch der Erfolg von Micropayment hat der Situation einen echten Sinn verliehen. Seit wir virtuelle Waffen für bares Geld verkaufen, gilt tatsächlich: Viele Waffen bringen auch viel Geld!

In diesem Workshop werden Sie eine möglichst große Anzahl von Schwertern in möglichst kurzer Zeit generieren. Die technische Seite ist dabei relativ einfach, schwieriger ist es, die generierten Schwerter nicht generisch wirken zu lassen. Die folgenden Beispiele wurden allesamt mit Maya erstellt. Sie können jedoch mit jedem 3D-Paket arbeiten, das die Möglichkeit zum Skripten bietet.

Je nachdem, ob Sie für ein 2D-Spiel oder für einen komplexen 3D-Titel arbeiten, werden Ihnen bestimmte Teile des Tutorials wichtiger sein.

Die Grundlagen

Wie schon in meinem letzten Workshop über automatische Content-Generierung in Ausgabe 01/2011 entsteht ein Teil der Vielfalt mittels Durchtauschen. Doch Vorsicht: Natürlich können Sie aus 10 Klingen, 10 Griffen und 10 Farben exakt 1.000 Kombinationen erstellen. Besonders abwechslungsreich wären diese allerdings nicht. Deshalb werden wir etwas mehr Handarbeit investieren müssen.

Stunde 1: Bildrecherche

Bei der Bildrecherche im Internet lohnt es sich, auch die zugehörigen Texte zu beachten. Sobald Ihnen ein interessanter Begriff auffällt, können Sie ihn für eine weitergehende Suche nutzen.

Fachbegriffe wie »Fehlschärfe« oder »Kopesch« liefern Ihnen auf Dauer vielfältigere Ergebnisse als die zwanzigste Seite zum Begriff »Schwert«.

Parallax Occlusion Mapping

Für die aktuelle Version der CryEngine 3 haben wir Parallax Occlusion Mapping (POM) integriert, um Macro Details hinzuzufügen, die nicht effizient mit Hardware Tessellation möglich wären.

Mit dieser Technik entsteht kein wirkliches geometrisches Displacement, stattdessen wird nur eine spürbare, blickabhängige Parallax-Bewegung vollzogen. Obwohl POM nicht so flexibel ist wie Hardware Tessellation, hat es einige wichtige Vorteile wie die Simulation von Macro Details und das Self-Shadowing.

Stunde 2 und 3: Modellieren der ersten Klingen

Ob Sie tatsächlich zwei Stunden für das Modellieren von mehreren Klingen brauchen, hängt von Ihrem Projekt ab. Low-Poly-Modelle gehen natürlich schneller von der Hand, Modelle für Großaufnahmen (wie die Bilder für diesen Artikel) dauern etwas länger. Folgende Punkte gilt es zu beachten:

- Wählen Sie extreme Modelle. Natürlich erkennt man eine auffällige Klinge in Kombination mit verschiedenen Griffen und Texturen schnell wieder. Doch für 1.000 unterschiedliche Schwerter brauchen wir auch extreme Unterschiede. Um das Problem mit dem Wiedererkennungseffekt kümmern wir uns später.
- Der Ansatz, mit dem die Klinge im Griff steckt, sollte eine gewisse Breite nicht überschreiten. Nicht jeder Ihrer Griffe wird breit sein, und auch breite Klingen sollten schließlich in jeden Griff hineinpassen (**Abbildung 1**).
- Achten Sie beim Modellieren auf das »Gewicht« der Klinge (**Abbildung 2**). Eine kurze Dolchklinge auf dem Griff eines Zweihänders würde im finalen Zustand absurd aussehen. Ihre Schwerter müssen nicht gleich lang sein, aber sie sollten in etwa gleich schwer sein.



- Modellieren Sie keine Gravuren, aber definieren Sie Bereiche, in denen solche passend wären.
- Stellen, die aus unterschiedlichen Materialien bestehen sollen, teilen Sie am besten in mehrere Objekte auf. Ordnen Sie diese stets als Klinge mit Unterobjekten an (Abbildung 3). Später beim Skripten können Sie ohne Aufwand die gesamte Objektgruppe auf einmal auswählen.
- Erstellen Sie zunächst noch keine UV-Maps. Die Klinge definiert die Form eines Schwertes am stärksten, gleichzeitig ist sie am einfachsten zu modellieren. Erstellen Sie daher ruhig einige Klingen mit ausgefallenen Varianten mehr.

Stunde 4: Modellieren der Griffe

Generell gilt das Gleiche wie bei den Klingen. Erstellen Sie zunächst einige extrem unterschiedliche Modelle. Halten Sie die eigentliche Grifffläche so einfach wie möglich. Mit Hilfe von Displacement Maps können Sie die Geometrie dieser Fläche unabhängig vom restlichen Griff durchtauschen.

Springen Sie zwischendurch zum nächsten Schritt und erstellen Sie zum Testen eine Displacement Map. Eventuell müssen Sie die Grifffläche aufgrund des Offsets dünner oder dicker anlegen als gedacht.

Stunde 5: Griff-Texturen und erste Metallfarben

Für den Anfang genügt es, ein Schwert in zwei Oberflächenarten zu unterteilen, in die Metall-

flächen und den gepolsterten Teil des Griffs. Natürlich haben echte Schwerter oft mehr als zwei Oberflächen. Es gibt beispielsweise stählerne Klingen auf einer vergoldeten Parierstange. Doch wenn Sie das Zusammenstecken von Griff und Klinge etwas kaschieren wollen, sollten Sie für beide dasselbe Metall verwenden.

Von Schwert zu Schwert darf (und sollte) sich das Material natürlich verändern. Erstellen Sie also zwei Materialgruppen: Metalle (Stahl, Eisen, Kupfer usw.) und Griffpolsterungen (etwa Leder und Stoffpolsterungen).

Verzichten Sie bei den Metallmaterialien noch auf alles, was von der UV-Map abhängt. Verteilen Sie also Farben und Reflexionswerte, verzichten Sie aber noch auf Texturen. Bei den Griffen hingegen ist bereits alles möglich.

Je nach Natur Ihres Projekts können andere Gruppen nötig werden. Es wäre etwa eine für Holzmaterialien nötig, um eine große Auswahl an Äxten oder Speeren zu generieren. Andere Projekte benötigen eventuell glühende Materialien für Magie- oder Laserschwert-Effekte. Versuchen Sie in diesen Fällen, die Anzahl der Materialien auf jeder Waffe gering zu halten. Bemühen Sie sich außerdem, besonders dominante Materialien auf allen Teilen der Waffe zu wiederholen.

Displacement Map oder doch nur Normal Map?

Auf Normal oder Displacement Maps sollten Sie auf keinen Fall verzichten. Besonders an der



Abbildung 1: Breite Klingen werden im Ansatz gestaucht, damit Sie auch auf schmale Griffe passen.



Abbildung 2: Die rechte Klinge gehört zu den längsten, bleibt aber dünn, damit sie nicht zu »schwer« wirkt.

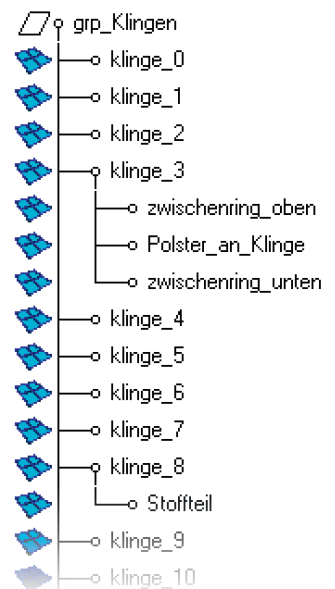


Abbildung 3: Klingen mit mehreren Materialien sollten stets hierarchisch angelegt werden.

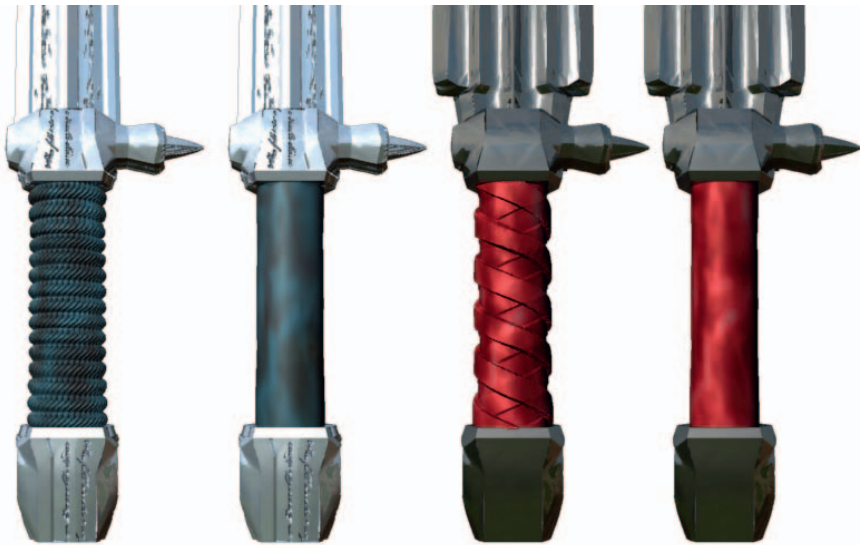


Abbildung 4: Griffstücke mit und ohne Displacement Map für die Veränderung der Geometrie.

Klinge haben Sie sonst kaum Möglichkeiten, die Geometrie durch abwechslungsreiche Texturen zu verändern. Vielfarbige Metalle gibt es schließlich selten, nur Specular und Reflection Map könnten noch komplexere Muster abbilden.

In vielen Fällen haben Sie keine Wahl: Manche 3D-Engines bieten keine Möglichkeit, mit Displacement Maps zu arbeiten. Sofern Sie die Möglichkeit haben, sollten Sie statt der Normal Maps lieber Displacement Maps wählen. Das liegt weniger an den Klingen als am Griff: Displacement Maps verändern auch die Silhouette

中易中标电子信息技术有限公司 曲线字库样品
中国标准技术开发公司

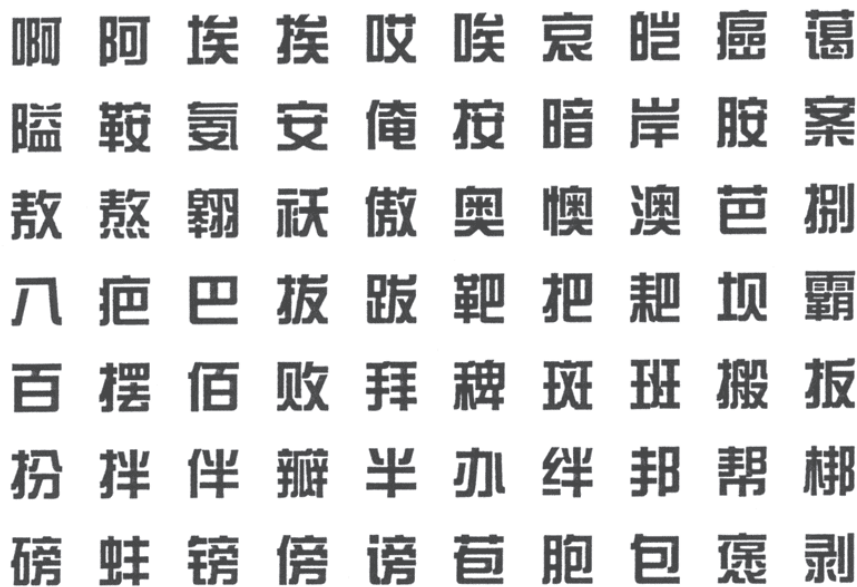


Abbildung 5: Mit der Schriftsammlung Ihres Vertrauens sind ein paar Gravuren schnell beisammen. Asiatische Schriften sollten vorher vom Übersetzer geprüft werden: Nicht nur eine ungeahnte Übersetzung kann Probleme bereiten – vermutlich wirkt der taiwanesisische Zeichensatz für Festlandchinesen so wenig mittelalterlich wie beispielsweise Arial oder Comic Sans für uns Europäer.

der Objekte (Abbildung 4). So scheinen dann sogar identische Griffstücke noch Unterschiede in der Form zu haben.

Eventuell hat der Spieler aber nie die Möglichkeit, Griffstücke der Schwerter nah genug zu bewundern – insbesondere wenn keine Nahansicht für das Inventarsystem integriert ist. In diesem Fall können Sie sich natürlich den ganzen Aufwand sparen.

Egal ob Sie mit Normal oder Displacement Maps arbeiten: Testen Sie diese möglichst früh in Ihrer Engine! In den meisten Fällen weicht die Darstellung stark von der ab, die Sie in Ihrer 3D-Software sehen.

Stunde 6 bis 8: Gravuren und die UV-Maps der Klingen

Klingen haben Gravuren. Elfishche Inschriften, keltische Knoten, dämonische Fratzen, geritzte Flüche, Engelsfiguren oder Totenköpfe.

Sie wollen solche Texturen für jede Klinge nutzbar machen? Wenn Sie die Texturen auf die Klingen anpassen, müssten Sie bei nur 5 Gravur-Texturen und 12 Klingen ganze 60 Texturen erstellen. Wesentlich effizienter: Sie erschaffen für jede der 12 Klingen je ein UV-Set, das an die Platzaufteilung der Gravuren angepasst ist. Vergessen Sie dabei nicht, dass UVs auch über die Texturen hinausragen dürfen. Um allerdings Kratzer, Rost oder getrocknete Blutflecken darzustellen, sind diese UV-Sets ungeeignet.

Falls Sie in Ihrem jeweiligen 3D-Programm ausgesprochen fit sind, empfehle ich multiple UV-Maps. Eine für die Gravuren, eine zweite für oben genannte Verschmutzungen. Je nach Geschmack können auch weitere UV-Maps erstellt werden, um etwa gefalteten Stahl darzustellen.

Wenn Sie unter Zeitdruck stehen oder falls Sie noch keine tiefgehende Erfahrung mit Ihrem 3D-Programm haben, können Sie diesen Schritt einfach überspringen. Das gilt besonders, wenn Sie für einen 3D-Titel arbeiten. Die auswechselbaren Texturen auf verschiedenen UV-Maps verschönern die Schwerter deutlich, müssen jedoch aufwendig auf eine einzige UV-Map gebakt werden, wenn es an den Export geht. Fragen Sie allerdings vorher Ihren Programmierer. Mit sehr viel Glück stört ihn eine zweite UV-Map überhaupt nicht und Sie können auf das Baken verzichten.

Stunde 9 bis 12: Varianten von Klingen und Griffen

Wenn die Texturen getestet und alle UV-Maps angelegt sind, erstellen Sie Varianten der Griffen und Klingen. Konzentrieren Sie sich zunächst auf die Klingen, die Silhouette der Klinge dominiert schließlich das gesamte Schwert.

Wenn Sie die Möglichkeiten Ihres 3D-Programms voll ausschöpfen, können Sie manche Variationen in weniger als einer Minute erstel-



len. Verlängern und verkürzen Sie Fehlschärfen. Geben Sie den Klingen eine konkave, eine konvexe und eine gerade Silhouette. Verbreitern Sie Klingen an Spitze oder Basis. Wichtig ist, dass sich die Silhouette verändert.

Die UV-Maps für Gravuren können Sie unverändert lassen, die UV-Maps für Verschmutzungen können Sie entweder neu erstellen oder die geringen Verzerrungen in Kauf nehmen.

In diesem Schritt löst sich auch das Problem mit den »zu einzigartigen« Klingen. Wenn Sie nur eine einzige Kopesch-Klinge haben, fällt diese dem Spieler als einzigartiges, schlecht integriertes Element auf. Wenn Sie vier unterschiedliche Varianten von Kopesch-Klingen modellieren, erkennt der Spieler ein Schwert zwar als Kopesch, hält die Varianten des Schwerts jedoch für einen sinnvollen Aspekt der Spielwelt. Dasselbe gilt im Übrigen für Griffe: Ein einzelner Griff mit auffallendem Drachenkopf ist ein Problem. Drei Griffe mit auffallenden Drachenköpfen repräsentieren eine Kultur.

Wenn Sie die Technik mit diesem Workshop nur ausprobieren, sollten Sie am Ende zumindest 13 Klingen haben. Für echte Projekte empfehle ich aber 23 oder mehr Varianten. Griffe benötigen Sie nicht ganz so viele, 11 Variationen sollten zum Testen ausreichen.

Puristen möchten vielleicht auch das Erstellen der Klingenvarianten automatisieren. Kunstgriffe wie Blendshapes, automatisierte Latices oder gar skriptgenerierte Stacheln und Klingenswellen sind natürlich möglich. Aber ganz ehrlich: In dieser Situation liefert Handarbeit immer noch das beste Ergebnis.

Stunde 13 bis 15: Skripten für den Export

Im Workshop zu Beginn von 2011 ging es ausschließlich darum, Bilder von den generierten Kombinationen zu rendern. Wenn Sie lediglich Bilder benötigen, können Sie tatsächlich auf dieselbe Methode wie in Ausgabe 01/2011 beschreiben zurückgreifen. In vielen Fällen werden jedoch exportierte 3D-Modelle benötigt. Und spätestens hier kommen Sie um das Skripten nicht mehr herum. Die nötigen Befehle lassen sich jedoch schnell recherchieren. Sie müssen wissen, wie eine for-Schleife funktioniert, wie Sie Objekte selektieren, wie sie die Materialien durchtauschen können und wie Sie über einen Skriptbefehl Ihren Exporter starten.

Dieser letzte Schritt ist wohl der aufwendigste in der Prozedur: Denn bei normalen Exporten müssen je nach Projekt, Exporter und Engine unterschiedliche Optionen gewählt werden. Recherchieren Sie daher die nötigen Befehle und stellen Sie mit ein paar Tests sicher, dass die per Skript exportierten Objekte genauso fehlerfrei im Spiel ankommen wie jene, die per Hand exportiert wurden.

Systematisch jede mögliche Kombination speichern?

Am naheliegendsten sind mehrere ineinander verschachtelte for-Schleifen. Jede davon zählt ein Element des Schwertes hoch – so wird sorgfältig jede der möglichen Kombinationen zusammengestellt. Die Reihenfolge hat System:

```
Klinge 1 mit Griff 1 mit Metall 1 mit Grifffarbe 1,
Klinge 1 mit Griff 1 mit Metall 1 mit Grifffarbe 2,
Klinge 1 mit Griff 1 mit Metall 1 mit Grifffarbe 3,
Klinge 1 mit Griff 1 mit Metall 2 mit Grifffarbe 1,
... und so weiter.
```

Der Nachteil: Wenn Sie nur die ersten 300 Schwerter exportieren, haben Sie weniger Variation als möglich wäre. Dasselbe gilt für die angepeilten 1.000 Schwerter. Denn bei der riesigen Anzahl an Kombinationen sind es eben auch »nur die ersten« 1.000 Stück. Damit die Auswahl unserer Schwerter weniger homogen ausfällt, wählen wir eine andere Methode.

Maximale Abwechslung

Verwenden Sie die Modulo-Funktion, um in einer einzigen for-Schleife in jedem Schritt jedes einzelne Element zu wechseln. Die Grundkonstruktion sieht (als Pseudo-Code) wie folgt aus:

```
for(int i = 0; i <
mengeDerZuErstellendenSchwerter; i++)
{
    leere die Auswahl;
    wähle Klinge Nr (i %
klingenAnzahl);
    wähle Griff Nr (i % griffAnzahl);
    weise Metall Nr (i %
metallAnzahl) zu;
    weise GriffMaterial Nr (i %
griffMaterialAnzahl) zu;
    Exportiere die Auswahl;
}
```

Anstelle (oder zusätzlich zu) der Export-Anweisung können sie Bilder natürlich auch rendern lassen, ganz nach den Anforderungen Ihres Projekts.

Um die gleiche Konstruktion konkret in Maya Skriptsprache MEL umsetzen zu können, müssen Sie einige Kleinigkeiten bei den Texturen vorbereiten. Erstellen Sie zunächst je ein Platzhaltermaterial für Metall und für Griffpolster. Weisen Sie die beiden Platzhalter den jeweiligen Polygonflächen zu und benennen Sie die Shading-Groups (nicht die Materialien selbst) »metal_SG« und »griff_SG«. Ihre bestehenden Materialien benennen Sie »metallMaterial_o«, »metallMaterial_1« usw.; die für die Griffpolsterung »griffMaterial_o«. Die Displacement-Materialien werden ab »metallDisplacement_o« bzw. »griffDisplacement_o« nummeriert. Das Skript in **Abbildung 6** geht davon aus, dass zu jedem Griffmaterial genau eine

```
int
$mengeDerZuErstellendenSchwerter =
1000;
int $klingenAnzahl = 17;
int $griffAnzahl = 11;
int $metallMaterialAnzahl = 9;
int $griffMaterialAnzahl= 5;

int $i;
for($i = 0; $i <
$mengeDerZuErstellendenSchwerter;
$i++){
    select -clear;
    select -add („klinge_“ + ($i %
$klingenAnzahl));
    select -add („griff_“ + ($i %
$griffAnzahl));

    connectAttr -force
(„metallMaterial_“ + ($i %
$metallMaterialAnzahl)+
„.outColor“) metall_SG.
surfaceShader;
    connectAttr -force
(„metallDisplacement_“ + ($i %
$metallMaterialAnzahl)+
„.outDisplacement“) metall_SG.
displacementShader;
    connectAttr -force
(„griffMaterial_“ + ($i %
$griffMaterialAnzahl)+
„.outColor“) griff_SG.surfaceShader;
    connectAttr -force
(„griffDisplacement_“ + ($i %
$griffMaterialAnzahl)+
„.outDisplacement“) griff_SG.
displacementShader;

    file
    -options „groups=0;ptgroup
s=0;materials=1;smoothing=1;norma
ls=1“
    -type „OBJexport“
    -exportSelected („C:/
Users/JW/Documents/maya/projects/
Artikel_Schwerter/data/Schwert_“ +
$i)
    ;
}
```

Abbildung 6: Das Skript geht davon aus, dass zu jedem Griffmaterial genau eine Displacement Map gehört.

Was ist Modulo?

Erinnern Sie sich noch an das Teilen in der Grundschule? Als das Ergebnis von »11 : 3« noch »3, Rest 2« war? Weil die Schüler noch keine Nachkommastellen kennen, bleibt beim Teilen oft ein Rest übrig. Modulo ist eben dieser Rest. »11 Modulo 3« ergibt 2, »3 Modulo 2« ergibt 1, »4 Modulo 2« ergibt 0. In den meisten Programmiersprachen wird Modulo mit dem »%«-Zeichen abgekürzt. Die Schreibweise ist also beispielsweise »11 % 3«. Wofür wird Modulo in unserem Fall verwendet? Wenn Sie etwa 13 Klingen modelliert haben und das 14. Schwert zusammensetzen, werden Sie bei den Klingen wieder von vorn beginnen wollen. Das Modulo-Zeichen erspart ihnen dabei einige Hilfskonstruktionen. Wenn Sie dem Schwert 134 beispielsweise die Klinge 134 % 13 geben, entspricht das der Klinge Nummer 4. Doch Vorsicht: Das Schwert 130 hat die Klinge »Nummer 0«, denn 130 ist glatt (und ohne Rest) durch 13 teilbar.

ne einzige Displacement Map gehört, dass sich jedoch die Metall-Displacements (also die Gravuren) frei mit den unterschiedlichen Metallen kombinieren lassen.

Sich häufig wiederholende Kombinationen

Möglicherweise fällt Ihnen beim Exportieren oder Rendern auf, dass bestimmte Elemente immer nur miteinander auftauchen. Vielleicht kommt die lange Klinge immer nur in Gold vor oder sie ist ausschließlich mit einem von drei Griffen kombiniert, nie aber mit einem anderen. Dieses Problem entsteht, wenn die Stückzahlen zweier unterschiedlicher Elementgruppen einen gemeinsamen Teiler haben. Wenn Sie etwa 12 Klingen und 8 Griffe (gemeinsamer Teiler 4) haben, wird die Klinge Nummer 1 nur in Kombination mit den Griffen 1 oder 5 auftreten. Noch schlimmer wird es, wenn zwei Elementgruppen exakt dieselbe Stückzahl haben. Es gibt zwei Wege, dieses Problem zu vermeiden. Erschaffen Sie einige einfache Varianten, sodass Sie auf Zahlen ohne gemeinsamen Teiler kommen. Oder ersetzen Sie den Term »(\$i % \$elementAnzahl)« eines einzelnen Elements durch die Zufallsfunktion »(rand \$i)«.

Fortgeschrittenen empfehle ich, eine »vorhersehbare« Zufallsfunktion zu nutzen. Diese wird allerdings in Maya schnell unübersichtlich. Der Modulo eines MD5-Wertes kann grundsätzlich genauso eingesetzt werden, ist aber nicht in jeder Skriptsprache verfügbar.

Dateinamen und Zusatzinformationen

Oftmals reicht es nicht aus, einfach nur Hunderte von Objekten zu speichern. So könnten die Spielwerte oder der Name des Schwertes mit der Grafik zusammenhängen. Ein Beispiel: Es gibt in Ihrem Spiel Kälteschaden und Sie haben dafür extra eine Eistextur unter die Metalle gemischt. Demzufolge sollte jedes Schwert mit dieser Textur den Namenszusatz »der Kälte« tragen. Es könnte auch sein, dass jedes Schwert mit Klinge 3, 8 oder 15 besonders viel Schaden anrichtet. Eventuell stoßen Sie auch auf das handfeste technische Problem, dass das Objektformat Ihrer Engine nicht die Verknüpfungen zu den richtigen Texturen speichert.

In jedem Fall müssen Sie zusätzliche Informationen speichern. Dabei gibt es wieder grundsätzlich zwei Möglichkeiten:

- Sie speichern alle Informationen (inklusive der verwendeten Einzelteile) in einer zusätzlichen Datei, beispielsweise als einfachen Text oder im XML-Format. Den Dateinamen der Objekte können Sie dabei einfach hochzählen, etwa »Schwert_0325.obj«. Vielleicht freut das Ihren Programmierer, weil er dann alle Schwerter mühelos einlesen kann. Es ist für Programmierer normalerweise einfacher, wenn nicht jedes Schwert eine eigene Informationsdatei hat, sondern wenn die Infos aller Schwerter in einer einzigen Datei zusammengefasst sind.

- Sie speichern die Informationen im Dateinamen. Es ergibt wenig Sinn, den String »... der Kälte« oder den Schadenswert in den Namen zu schreiben. Besser ist es, durch Zahlenwerte direkt die verwendeten Einzelteile aufzuzählen. Zusatzinformationen wie etwa der Name oder die Schadenswerte werden erst später im Programmcode hinzugefügt.

- Verwenden Sie bei dieser Methode Buchstabenkürzel. Wenn Sie aus 1.000 Dateien namens »Schwert_18_4_7_11.obj« alle Dateien mit dem Metallmaterial 9 herausuchen wollen, stehen Sie vor einem Problem. Sofern Sie Ihre Dateien »Schwert_k18_g4_m7_s11.obj« benannt haben, müssen Sie Windows nur nach dem Dateinamen »*_m9_*« suchen lassen, um alle Objekte mit dem Metallmaterial 9 zu finden.

Stunde 16: Schönheitskorrekturen und Nachbessern

Ihnen sind mittlerweile sicherlich Dutzende Kleinigkeiten aufgefallen, die Sie noch ändern wollen. Passen manche Texturen einfach nicht? Lassen sich möglicherweise manche Klingen stilistisch nicht mit anderen Griffen kombinieren? Ist noch mehr Abwechslung bei den Klingen nötig? Oder wünschen Sie sich sogar ein neues Element wie etwa durchtauschbare Stacheln?

Wenn Sie diesen Workshop das allererste Mal durchgearbeitet haben, werden Ihre Ergebnisse noch nicht so gut aussehen wie Ihre »normalen« Werke. Schmeißen Sie nicht das Handtuch! Denn jede Verbesserung, die Sie jetzt einbauen, betrifft nicht nur ein Schwert – sie betrifft Hunderte!

Jochen Winkler