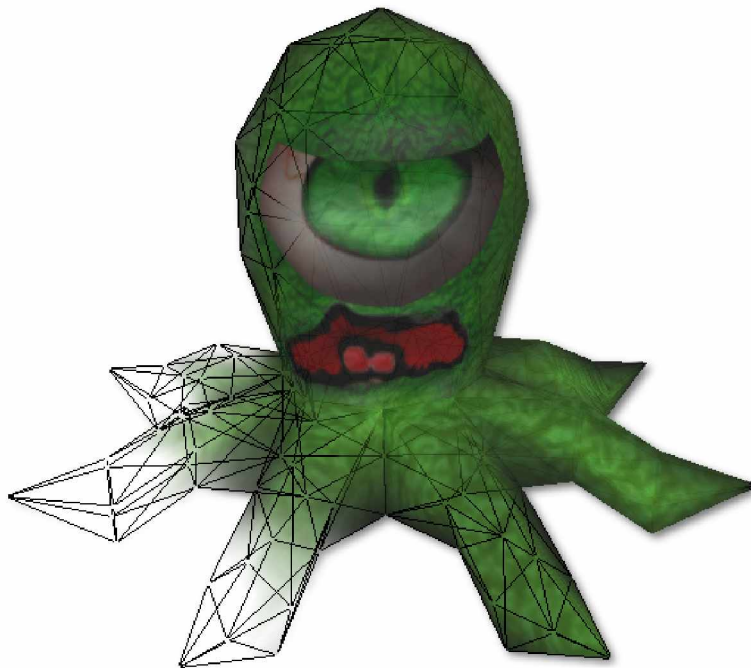




Einführung in 3D-Dateien am Beispiel der Software MilkShape3D



Erstellt am
13.01.05

von
Ilja Herlein

<http://www.iljaherlein.de>
ilja@itecnix.com

**1. Einleitung**

- a. Die virtuelle Welt im 21sten Jahrhundert 3
- b. Was ist MilkShape3D? 3

2. 3D-Grundlagenwissen

- a. Vertices 5
- b. Faces 5
- c. Objekte 6
- d. Gruppen 6
- e. Materials / Skins 7
- f. Joints & Skeleton 7

3. MilkShape3D Benutzeroberfläche

- a. Ansichtsfenster 8
- b. Model 9
- c. Groups 10
- d. Materials
 - 1. Texturen 11
 - 2. Farben 11
 - 3. Deckkraft / Transparenzen 12
 - 4. SphereMaps 13
- e. Joints 13
- f. Wichtige Tastenkürzel 14

4. Erstellung eines Models

- a. Vorarbeit 16
- b. Modelling (in 10 Schritten) 18
- c. Skinning 25

5. Quellen 28



1 Einleitung

1.a Die virtuelle Welt im 21sten Jahrhundert

In der Welt des 21sten Jahrhundert beherrscht die virtuelle Welt unseren Alltag so sehr wie nie zuvor. Fernsehen, Videospiele, ein virtueller Spaziergang durch eine fremde Stadt oder gar ein hochkomplexer Flugsimulator zur Ausbildung von Piloten.



Wir befinden uns an einem Punkt, an dem es uns sowohl hardware- als auch softwaretechnisch möglich ist, 3D Welten zu erschaffen, die dem Benutzer einen hohen Grad an Realität und Unterhaltung bieten können.

Dazu wird allerdings fundiertes Grundlagenwissen benötigt. Man muss sowohl den Aufbau als auch die Funktionsweise von 3D Objekten begreifen, verinnerlichen und anwenden können.

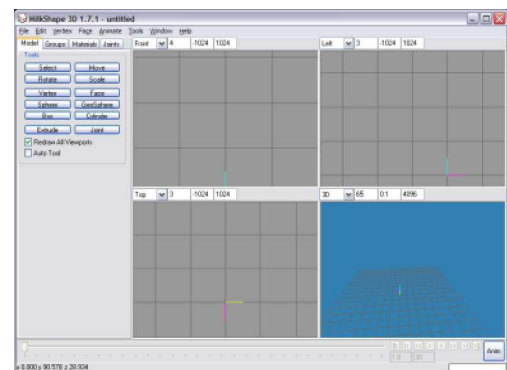
Ich werde in dieser Ausarbeitung die Grundlagen des 3D-Modelling, Texturing sowie Animating erläutern. Zwar wird man nach dem konzentrierten und interessierten Lesen dieser Lektüre kein Meister im 3D-Bereich sein, jedoch wird man mit Begriffen wie Vertex, Polygon oder Joint etwas anfangen und bereits einfache Modelle erstellen können.

1.b Was ist MilkShape3D?



Als im Jahre 1998 der Ego-Shooter „Half-Life“ von Valve erschien, gewann das Spiel sehr schnell eine große Fan-Gemeinde, die es um weitere Charaktere und Levels noch weiter ausbaue wollte. Die wenigsten Spieler waren auch gleichzeitig professionelle 3D-Entwickler und so war die Nachfrage nach einem relativ einfach zu bedienenden 3D Editor groß.

Ein talentierter und engagierter Schweizer Programmierer, Mete Ciragan, machte sich damals daran, ein relativ simples, jedoch für die Zwecke vollkommen ausreichendes Programm zum Modellieren neuer Figuren und Objekte zu entwickeln. Das Resultat, „MilkShape3D“, fand in der Fangemeinde von Half-Life großen Anklang und





Gefallen. Schon bald war klar, dass diese Software sehr viel mehr Potential beinhaltet, als nur für ein bestimmtes Spiel zu bestehen. Aus diesem Grunde entschied sich Mete Ciragan, die Software kommerziell als eigenständiges Produkt weiter zu entwickeln und sie somit für den gesamten 3D Markt interessant zu machen. Mittlerweile beherrscht MilkShape3D offiziell 37 Import- und Ausgabeformate und ist sowohl auf viele beliebte Spiele (Half-Life, Unreal, Die Sims, ...) als auch auf viele bekannte große 3D-Programme (3D Studio Max, Maya, Lightwave, ...) abgestimmt. Durch die enge Zusammenarbeit mit der Fangemeinde und dem speziell angefertigten SDK (Software Development Kit) wurde die Möglichkeit der Entwicklung von neuen Plug-Ins (Programmerweiterungen) und Dateiformaten (Ex- und Import) geschaffen.

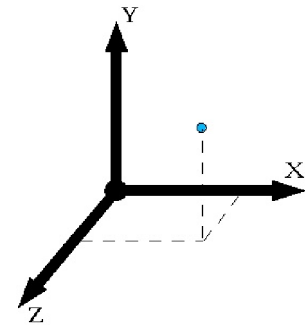
Milkshape3D ist für alle geeignet, die neu ins 3D-Design einsteigen möchten, weil es auf die wesentlichen Merkmale und Anforderungen an ein 3D Programm reduziert ist, sich dabei sehr gut und einfach bedienen lässt und zudem kompatibel zu fast allen anderen 3D Programmen ist. Aber auch für fortgeschrittene und professionelle 3D Designer ist Milkshape eine sehr gute Wahl, wenn es um



2. 3D-Grundlagenwissen

2.a Vertices

Möchte man zum Beispiel zu Statistik-Zwecken ein zwei-dimensionales Diagramm erstellen, so kann man dies durch Angabe von zwei Koordinaten tun, die jedem Koordinatenpunkt eine eindeutige Position zuweisen. Analog dazu funktioniert dies bei 3D Koordinaten. Jeder Punkt im Raum ist durch 3 Koordinatenangaben (x-, y- & z-Achse) bestimmt. Ein solcher Punkt wird **Vertex** (Plural: **Vertices**) genannt

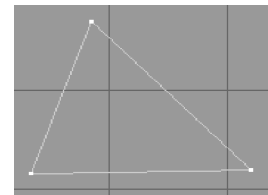


Diese Koordinaten können entweder per direkter Werteingabe erfolgen oder aber, wie es im Model-Design für Spiele eher üblich ist, per Interaktion in einer auf 2D reduzierten Ansicht. Exakte Koordinateneingaben sind normalerweise nur in CAD-Anwendungen (*Computer Aided Design*) wie „Auto CAD“ erforderlich, da diese zum Konstruieren im Ingenieurwesen benutzt werden, wo es auf Bruchteile von Millimetern ankommen kann.

2.b Faces

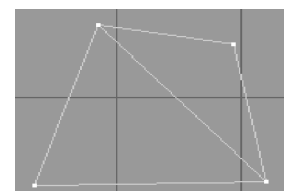
Da Vertices zwar exakte Punkte im Raum sind, jedoch keine Größe besitzen, sind sie nicht sichtbar und reichen deshalb alleine nicht aus, um 3D-Modelle darzustellen. Aus diesem Grund benötigt man erkennbare Flächen, die durch Vertices als Eckpunkte definiert werden.

Faces werden in MilkShape von genau 3 Vertices bestimmt, da man im Low-Poly Bereich (z.B. 3D-Spiele) versucht, mit so wenig wie möglich Informationen, so viel wie möglich darzustellen und ein Dreieck die Figur mit den wenigsten Ecken ist. (Low-Poly = geringe Anzahl an Polygonen/Faces)



Allgemein werden solche sichtbaren Flächen *Polygone* genannt und sind, wie der Name bereits sagt (poly = viel | gon = Winkel/Ecke | Polygon = Vieleck) in der Anzahl ihrer Ecken nicht festgelegt. Je nach Absicht kann es von Vorteil sein, wenn die Flächen Vierecke sind, da sich dann z.B. die Schattierung ganz anders berechnen lässt.

Faces können sich ihre Eckpunkte untereinander teilen, sodass man z.B. mit 4 Vertices 2 Faces darstellen kann.

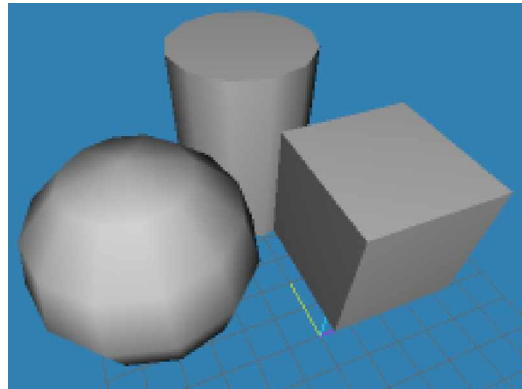




Da es sich hierbei um eine Fläche im Raum handelt ist sowohl eine Vorderseite als auch ein Rückseite vorhanden. Zur Darstellung der Rückseite gibt es zwei Möglichkeiten, die allein auf Render-Einstellungen basieren und deshalb frei wählbar sind (als **Rendern** bezeichnet man den Vorgang der Umsetzung der Informationen in ein sichtbares Bild): Bei der einen ist die Funktion „*Draw Backfaces*“ (erreichbar über die rechte Maustaste auf ein Ansichtsfenster) eingeschaltet; es wird dann auf der Rückseite spiegelverkehrt das gleiche dargestellt wie auf der Vorderseite. Im zweiten Fall ist die Funktion „*Draw Backfaces*“ ausgeschaltet; dann ist das Face nur von einer Seite (Vorderseite) sichtbar. Es hat dann quasi keine Rückseite mehr, sodass man von der anderen Seite durchschauen kann.

2.c Objekte

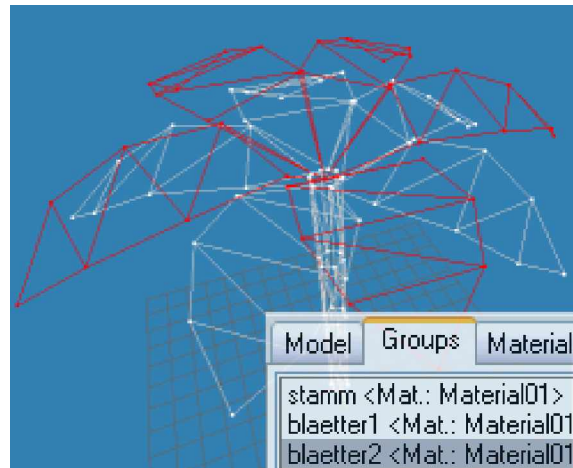
Objekte sind keine neuen 3D-Bestandteile, sondern bestehen aus den zwei zuvor behandelten Komponenten (Vertices & Faces). Sie sind lediglich vorgefertigte geometrische Figuren, die durch Eingabe gewisser Variablen beeinflusst werden können. Auf Grund ihres hohen Nutzens und der erheblichen Vereinfachung sind sie in nahezu jedem 3D-Programm zu finden. Sie bilden quasi einen



Ausgangspunkt bei erstellen neuer Gebilde, da sich fast alles vereinfacht als eine Zusammensetzung von Würfeln, Kugeln oder Zylindern darstellen lässt.

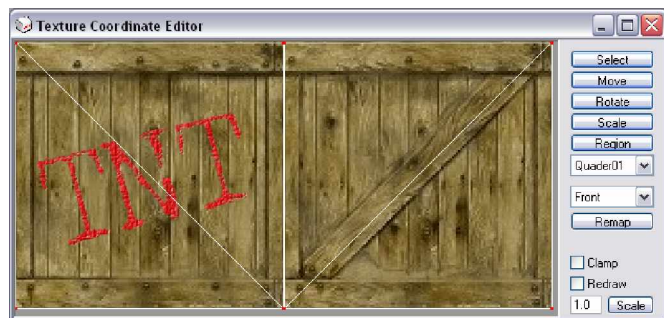
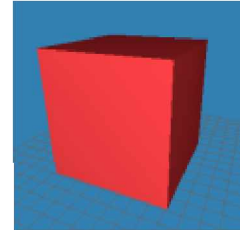
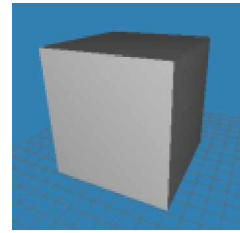
2.d Gruppen

Zur Erleichterung beim Modellieren lassen sich Faces zu beliebig großen Gruppen verbinden. Dieser Vorgang hat keinerlei Auswirkung auf das Aussehen des Models, sondern dient vielmehr dem Komfort beim Erstellen, aber auch bei der späteren Echtzeit-Darstellung, da man an dann gezielt auf einzelne Gruppen zugreifen kann. Im rechten Beispiel lassen sich die oberen Palmblätter über den Gruppennamen „blaetter2“ selektieren.



2.e Materials / Skins

Da man durch Faces lediglich Flächen definiert, diese jedoch keine weiteren Eigenschaften wie Farben oder gar Strukturen besitzen, muss man diese nachträglich festlegen. Dies geschieht durch *Materials*, oft sehr treffend auch als *Skins* (skin = Haut) bezeichnet, da man dem Objekt ja quasi eine Haut überzieht und erst dann das Aussehen erreicht, das man möchte. Diese Haut kann zum einen lediglich einen Farbton bekommen, was den Vorteil hat, dass man auf einfache Weise ein Objekt einfärben kann, ohne externe Dateien zu verwenden und die Farbe in Echtzeit manipulieren kann. So wäre es beispielsweise denkbar, einen Ball in einer Simulation oder einem Spiel stetig in Regenbogenübergängen zu verändern, indem man auf den Farbwert per Quellcode zugreift.

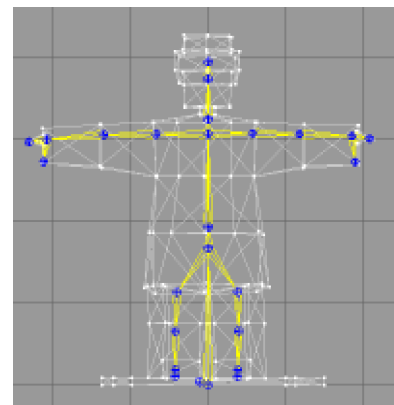


Zum anderen, und das ist die üblichere und meist sinnvollere Vorgehensweise, lassen sich 2D Grafiken in den Standard-Formaten wie JPG, BMP oder PNG auf das Model aufspannen. Dies kann, gekonnt eingesetzt, zu einem sehr professionellen und nahezu fotorealistischen Resultat führen, da man für diese Grafiken ebenfalls bearbeitete Fotografien verwenden kann.



2.f Joints / Skeleton

Um Objekte animieren zu können, verwendet man ein Skeleton, das schon vom Namen her die Funktionsweise erahnen lässt. Dieses Skelett besteht aus mehreren an einander hängenden *Joints* (joint = Gelenk, Verbindung), denen wiederum Vertices zugeordnet sind. Beim Animieren arbeitet man dann nur an den Joints, die dann die Position der Vertices und somit das Aussehen des Models verändern.





3. MilkShape3D Benutzeroberfläche

3.a Ansichtsfenster

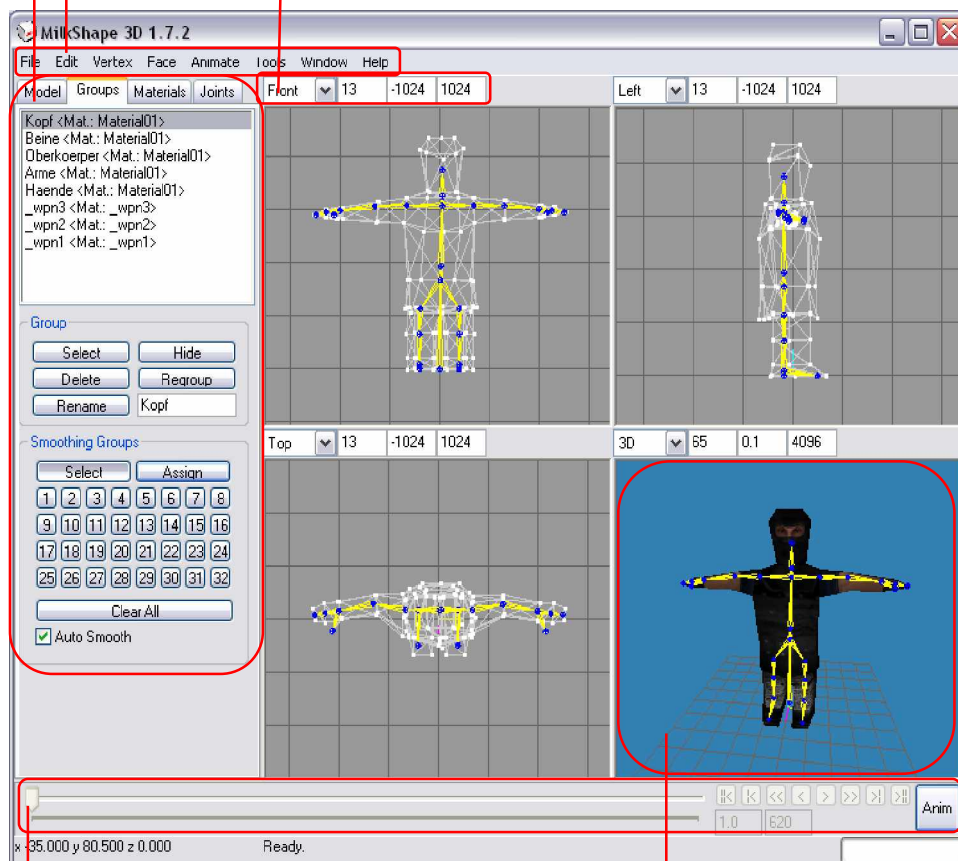
Zum Betrachten und Bearbeiten der 3D-Modelle dienen die 4 Ansichtsfenster, deren Darstellungsweise frei gewählt werden kann. Jedoch sind die Standardeinstellungen bereits sehr sinnvoll gewählt und können so beibehalten werden.

Zusätzlich sollten vor allem im Anfangsstatus die Fenster-Optionen aktiviert werden (*Window / Show Viewport Caption*) um zu jedem Zeitpunkt genau zu wissen aus welcher Perspektive man das Geschehen gerade beobachtet oder um gegebenenfalls auf eine andere Perspektive umschalten zu können.

Control Panel (Bedienfeld): In 4 ‚Karteikarten‘ finden sich hier thematisch gegliedert die wichtigsten Werkzeuge zum Erstellen & Bearbeiten von Model, Textur & Skelett

Menüleiste: Hier befinden sich alle erweiterten Werkzeuge, Zusatzmodule und Programmeinstellungen

Fenster-Optionen: Für Anfänger ist hier lediglich der erste Wert wichtig. Er gibt die aktuelle Sichtrichtung an. Die anderen 3 Werte stehen für Linsenwinkel und Sichtbegrenzungen und sollten nicht verändert werden.

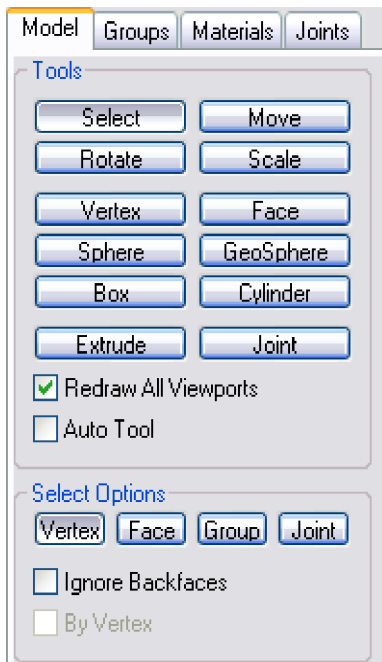


Die **Zeitachse** wird nur im Animationsmodus benötigt. Kurz zusammengefasst werden dabei Joints verschoben und ihre Positionen auf dem aktuellen Keyframe (Schlüsselbild) gespeichert. Bei der Wiedergabe der Animation werden dann die Zwischenstufen interpoliert/berechnet.

Jedes **Darstellungsfenster** lässt sich über einen Rechtsklick frei konfigurieren. Angefangen bei der Perspektive über Rendermodus bis hin zum Hintergrundbild



3.b Model



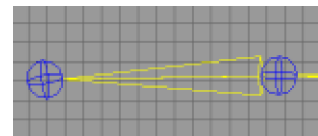
Die Model-Werkzeuge beinhalten alle notwendigen Arbeitsmittel zum Erstellen und Verändern von Objekten. Mit „**Select, Move, Rotate & Scale**“, die ebenfalls sehr komfortabel über die Tasten **F1 bis F4** erreichbar sind, lassen sich Vertices, Faces, Gruppen und Joints **auswählen, verschieben, drehen** und in der **Größe verändern**.

Vertex platziert neue Vertices an die Position der Maus, wobei die Tiefenachse der aktuellen Ansicht immer 0 sein wird, da man sonst die Position nicht kontrollieren könnte (in der Front-Ansicht wäre das z.B. die z-Achse). Mit Face kann man nun diese Vertices verbinden. Dazu klickt man hintereinander 3 Vertices gegen den Uhrzeigersinn an, um die Zugewandte Seite zur Vorderseite zu machen. Entsprechend erzeugt man gegen den Uhrzeigersinn eine Rückseite (Backface). Die Unterschiede wurden bereits bei den Grundlagen besprochen.

„**Sphere, Geosphere, Box & Cylinder**“ erzeugen die besprochenen Standard-Objekte Kugel (in zwei unterschiedlichen Aufbauformen), Quader und Zylinder.

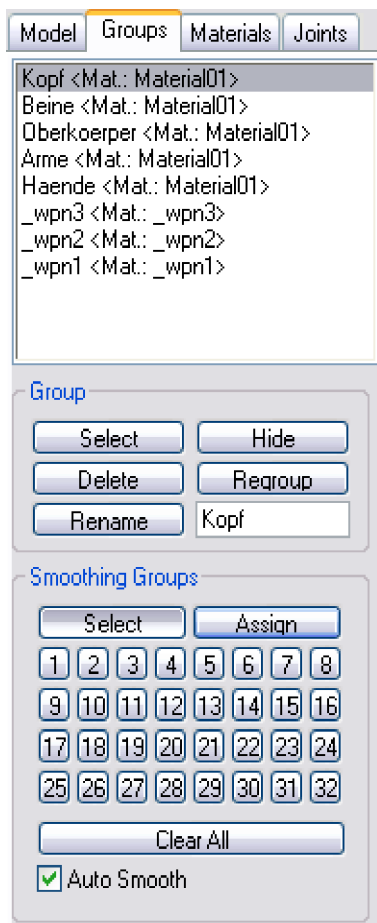
Extrude ist ein sehr wichtiges und oft genutztes Werkzeug. Es ermöglicht das extrudieren (herausheben) von ausgewählten Faces, wobei automatisch Verbindungswände zum Ursprungsobjekt entstehen.

Joint dient zum Erstellen von Skeletons und funktioniert auf ähnliche Weise wie das Vertex-Werkzeug. Ist beim Setzen ein anderer Joint selektiert, werden beide automatisch mit einander verbunden. Die Pfeilrichtung gibt an, welcher Joint welchem zugeordnet ist. Das ist in so fern wichtig, weil sich Veränderungen (wie z.B. Verschieben) am übergeordneten auf das untergeordnete ‚Gelenk‘ auswirken. Es handelt sich hierbei um eine Art Vererbung.





3.c Groups



Strukturierte und übersichtliche Gestaltung von Gruppen kann vor allem im fortgeschrittenen Verlauf sehr sinnvoll und zeitsparend sein. Wie im linken Beispiel wurden die einzelnen Körperteile zu separaten Gruppen zusammengefasst.

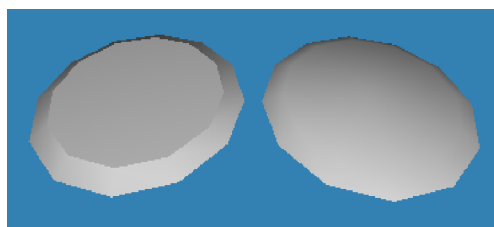
Je nach Bedarf und Komplexität des Models könnte auch dies noch verfeinert werden; so wären z.B. auch Unterteilungen in Hand_links und Hand_rechts oder auch Brust, Bauch, etc. denkbar.

Diese Gruppen werden zudem beim **Texturieren** benötigt, da man es dort mit immer nur einer Gruppe auf ein Mal zu tun hat.

In diesem Menü kann man die ausgewählte Gruppe selektieren, verstecken, löschen und umbenennen (**Select, Hide, Delete, Rename**).

Regroup erstellt aus den aktuell selektierten Faces eine neue Gruppe. Gehören diese Faces bereits einer Gruppe an, so werden sie von dort übernommen.

Smoothing Groups werden vom Laien gerne übergangen, weil sie scheinbar nichts Offensichtliches bewirken. Tatsächlich jedoch sind diese Gruppierungen sehr wichtig, da sie über die weiche Schattierung des Objekts Zusammengehörigkeiten und Rundungen (durch flüssige Übergänge), sowie Diskrepanzen und harte Kanten (durch abrupte Schattenintensitätslinien) ausdrücken können.



Die beiden Modelle in diesem Beispiel sind in ihrem Aufbau identisch und unterscheiden sich lediglich bei den Smoothing Groups.

Das linke Objekt besteht aus zwei, das rechte aus nur einer „Oberflächenglättungs-Gruppe“.



3.d Materials



1. Für ein 3D-Model können beliebig viele Grafik-Dateien als Textur/Skin verwendet werden. Diese werden im Materials-Bereich organisiert und angewandt.

Der **New** Button erzeugt intern einen neuen Eintrag für die Textur, die aber erst über **<none>** gewählt werden muss, sodass sich die Beschriftung des Buttons in den Dateinamen ändert. Über die obere Liste lassen sich alle vorhandenen Texturen wählen.

Zum Anwenden einer Textur auf selektierte Faces genügt ein Klick auf **Assign**.

Die genauere Platzierung der Grafik auf dem 3D-Model wird unter **4.c Skinning** erläutert.

2. Benötigt man keine Grafik als Oberflächenmaterial, so kann man auch Farben verwenden.

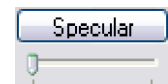
Ambient: Farbe, die über das Model gelegt wird; sowohl über bereits vorhandene Texturen oder Farben. (ambient = umgeben)

Diffuse: Farbe, die von der Oberfläche reflektiert wird. Da als Standardlicht fast immer unsichtbares bzw. weißes Licht verwendet wird, gibt dieser Farbton die eigentliche Farbgebung

des Objektes an.

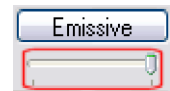
Specular: Gibt die Farbe an, mit der das Objekt imaginär bestrahlt wird.

Vorhandene gleiche Farben werden somit herausgehoben, da sie dann stärker leuchten als der Rest. Die Intensität kann über den Schieberegler darunter variiert werden.



Emissive: Definiert die Leuchtfarbe, die unabhängig von dem Einstrahlungslicht als oberste Farbschicht fungiert.

3. Der Schieberegler unter „Emissive“ legt die **Deckkraft** des Materials fest. Somit kann man stufenlose Transparenz festlegen, wobei ganz rechts 100% und ganz links 0% Deckkraft definiert sind. Jedoch ist hier Vorsicht geboten. Nicht immer wird man mit dem Ergebnis zufrieden sein, da Darstellungsfehler auftreten können. Dies ist darauf zurückzuführen, dass es im 3D Bereich immer auf die **Renderreihenfolge** ankommt und sich diese nicht immer (zufrieden stellend) beeinflussen lässt.



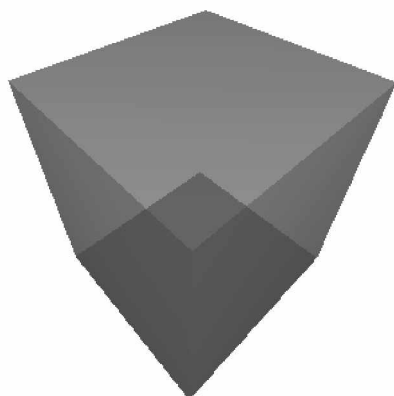
Zur Erklärung: Objekte werden immer als Ganzes in der Reihenfolge der



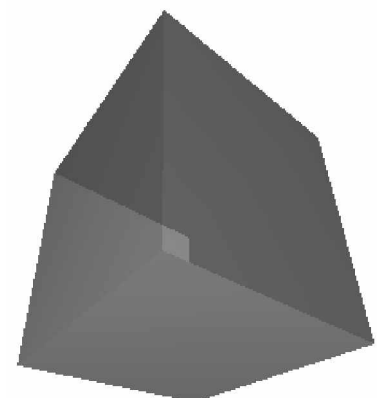
gespeicherten Koordinaten gerendert. Überlappen sich nun **mehrere Objekte** und ein Objekt mit einer Teiltransparenz, das sich räumlich gesehen vor einem anderen Objekt befindet, verdeckt das hintere Objekt, sodass man durch diese Teiltransparenz nur alles zeitlich vorher Gerenderte sehen kann.

Im linken Beispiel aus einem 3D-Spiel haben die Palmblätter einen weichen Rand um natürlicher zu wirken. Beim genaueren Hinsehen fällt jedoch auf, dass Teile des Palmstammes im Hintergrund nicht sichtbar sind, sondern stattdessen der Sand der Bodentextur zum Vorschein kommt. Die Palme im Hintergrund wurde also vor der Palme im Vordergrund gerendert.

Dieser Effekt kann aber auch bei einem **einzelnen Objekt** auftreten, wenn sich eigene Teile überlagern oder wenn man durch das Objekt komplett durchschauen kann.



In diesem Beispiel wurde ein normaler Würfel erstellt und seine Oberfläche auf 50% Transparenz gestellt. Während das Bild auf der linken Seite so dargestellt wird, wie man es möchte, kommt auf der rechten Seite genau das besprochene Problem zum

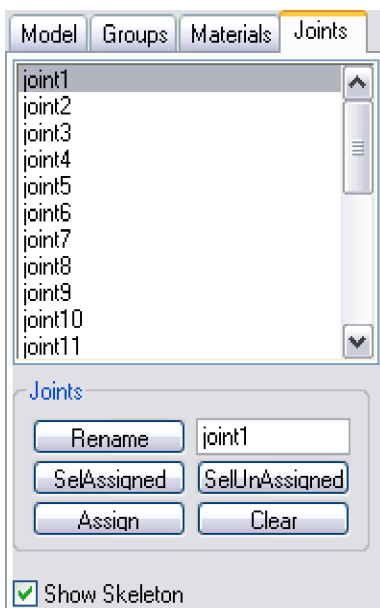


Vorschein. Die Unterseite des Würfels wird zuerst gerendert und ermöglicht deshalb kein Durchsehen, da die anderen Seiten zum Renderzeitpunkt noch nicht existieren.



4. Verwendet man eine Textur als **SphereMap**, so befindet sich diese ☒ SphereMap Grafik anschließend nicht fixiert auf dem Objekt sondern variiert je nach dessen Ausrichtung. Man kann sich die Funktionsweise bildlich relativ leicht vorstellen: Die Textur wird auf eine imaginäre, fixe Kugel um das Objekt herum gespannt und anschließend darauf projiziert. In der Praxis wirken die Objekte dadurch sehr metallisch, da sie ihre „Umgebung“ scheinbar widerspiegeln.

3.e Joints



Im Joints-Bereich werden alle erzeugten Joints verwaltet.

Es ist grundsätzlich besser die Joints über „**Rename**“ signifikant zu benennen, damit man bereits im Menü erkennen kann, wo sich welcher Joint befindet. Da man diese jedoch auch direkt in den Arbeitsfenstern selektieren kann, ist diese Benennung nicht zwingend erforderlich.

Essentiell hingegen ist die Möglichkeit, selektierte Vertices an den in der Liste gewählten Joint binden zu können („**Assign**“) oder wieder freizugeben („**Clear**“).

Zugewiesene Vertices lassen sich über die Auswahl des gewünschten Joints in der Liste und einen Klick auf

„**SelAssigned**“ selektieren.

Ungebundene Vertices kann man leicht über „**SelUnAssigned**“ feststellen. Je nach Bedarf lässt sich das „Skelett“ über „**Show Skeleton**“ ein oder ausblenden, um die Arbeitsfläche übersichtlicher darzustellen, wenn man es gerade benötigt.



3.f Wichtige Tastenkürzel

Nachfolgend sind die aus Erfahrung heraus am häufigsten verwendeten Shortcuts aufgeführt. Selbstverständlich gibt es noch weitere und bei Bedarf können über „**Tools** → **ShortCut & PlugIn Manager** ...“ eigene hinzugefügt und bearbeiten werden.


Möchte man die Nachfolgende Erstellung eines Models nacharbeiten, so empfiehlt es sich zuerst, sich mit dem Programm und seiner Bedienung auseinander zu setzen. Dazu gehört auch mit den Tastenkürzeln vertraut zu sein.

Tastatur:

F1, F2, F3, F4	Select, Move, Rotate, Scale (Aus dem Model-Tab) → Man sollte sich angewöhnen, beim Modellieren die linke Hand so auf die Tastatur zu legen, dass vier Finger der linken Hand auf diesen vier F-Tasten liegen und man so zwischen den Funktionen sehr schnell wechseln kann. Der Daumen kann die Leertaste bedienen. Selbstverständlich muss die Hand für andere wichtige Shortcuts bewegt werden, jedoch ist es wichtig diesen grundlegenden „Handgriff“ zu beherrschen, weil es sehr zeitsparend sein kann.
Strg+Z	Letzten Schritt rückgängig machen.
Strg+R	Rückgängig gemachten Schritt wiederholen
Strg+W	„Weld“ (=verschweißen) → Befinden sich unter der aktuellen Selektierung Vertices mit gleichen Koordinaten, so werden diese zu einem Punkt zusammengefasst. Dies ist vor allem dann wichtig, wenn man das Model aus mehreren Teilen zusammensetzen möchte.
Strg+N	„Snap Together“ (=zusammenschnappen) → Alle aktuell selektierten Vertices werden dadurch an dieselbe Stelle, die sich in deren Mitte befindet, verschoben. Meist ist im Anschluss der Befehl „Weld“ sinnvoll.
Leertaste	Vergrößert das aktuell aktive Fenster. (Rückgängig ebenfalls mit Leertaste)
Strg+D	Dupliziert die aktuell selektierten Faces. Diese bleiben an derselben Stelle und sollten deshalb gleich verschoben werden, da sich sonst unerwünschte Überlappungen ergeben können.
Strg+A	Selektiert alles Sichtbare.
Strg+Shift+A	Deselektiert alles.
Strg+I	Invertiert die Selektierung.
Strg+H	Versteckt alles Selektierte. Sinnvoll, wenn man einen Bereich bearbeiten möchte, der auf Grund seiner Komplexität nur schwer überschaubar ist.
Strg+Shift+H	Lässt alles Versteckte wieder erscheinen.
Entf	Löscht das aktuell Selektierte.
Strg+T	Öffnet den „ Texture Coordinate Editor “, der benötigt wird um Texturen auf einem Objekt richtig zu platzieren.



Maus: (Maustaste = MT)

Strg + Linke MT	Sichtbereich verschieben
Mausrad	Zoomen. Lässt sich durch Strg beschleunigen
Rechte MT	Menü für das jeweilige Fenster
Im -Modus:	
Linke MT	Klick: punktiert Selektieren. Klicken-Ziehen: Rahmen-Selektierung
Strg+ Rechte MT	Klick: punktiert Deselektieren. Klicken-Ziehen: Rahmen-Deselektierung

Tipp 1: Oftmals ist es nur sehr schwer möglich einige Vertices gezielt zu selektieren. Es ist dann unter Umständen sinnvoller, mehr als notwendig zu selektieren und das nicht benötigte (in einem oder mehreren Schritten) wieder zu deselektieren. Speziell hier ist die Verwendung von 3 Ansichtsfenstern wichtig, da man aus bestimmten Blickwinkeln besser interagieren kann.

Tipp 2: Was nur wenige wissen: Es ist auch möglich im 3D-Fenster zu selektieren. Dazu muss lediglich die **Alt**-Taste gedrückt werden, während man klickt oder einen (nicht sichtbaren Rahmen) zieht.

Das Selektieren von gewünschten Vertices oder Faces kann für einen Anfänger unter Umständen zum akrobatischen Akt werden. Ein guter Umgang mit einem Programm lässt jedoch sich nicht durch Lesen, sondern durch Ausprobieren, Üben und Dranbleiben erzielen.



4. Erstellung eines Models

4.a Vorarbeit

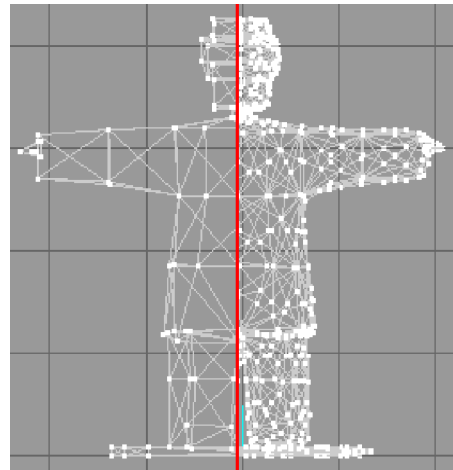
Bevor man das 3D-Programm überhaupt startet und wild drauf los designed, sollte etwas Vorarbeit geleistet werden.

Wichtig ist vor allem, einige Fragen im Vorfeld zu klären, um unnötige Probleme zu vermeiden:

- Low Poly oder High Poly?

Wird das Model später in Echtzeit gerendert (Spiel, Simulation) oder als Bild/Video vorgerendert?

Im ersten Fall ist es wichtig zu beachten, dass mit steigender Polygon-Zahl auch die Ansprüche an den Rechner des Endanwenders steigen, da mehr Rechenoperationen durchgeführt werden müssen. Somit wäre Low Poly empfehlenswerter. Wobei „Low“ (= gering) selbstverständlich relativ zu sehen ist. Werden nur wenige Objekte benötigt, können diese jeweils etwas mehr haben, als wenn es sich um viele Objekte handelt.



Im zweiten Fall spielt die Anzahl der Polygone eine eher zweitrangige Rolle, denn hier muss der Rendervorgang nur ein Mal beim Ersteller erfolgen. Der Endanwender hat es dann nur noch mit einer Bild- bzw. Videodatei zu tun, die keine Informationen über die ursprüngliche Anzahl der Polygone beinhaltet. Jedoch steigt mit der Komplexität des Models die Renderdauer, was sich bei sehr komplexen und langen Videosequenzen sehr viel ausmachen kann. Benötigt man z.B. 3 Minuten pro Bild würde das bei 25 Bildern pro Sekunde und einer Länge von einer Minute beachtliche 75 Stunden zum Rendern benötigen. Diese Rechenarbeit lässt sich zwar auf mehrere Rechner verteilen, oder im professionellen Bereich gar über Rechenzentren erledigen, allerdings ist die Zeit- und Kostenfrage nicht unerheblich.

Es gilt immer: **So viel wie nötig - so wenig wie möglich.**



- **Statisch oder animierbar?**

Muss ein Objekt später animiert werden, so sollte es in einer Ausgangslage erstellt werden, aus der man es ohne Probleme bewegen kann. Dient es nur als statisches Objekt, so kann es gleich in der der Endposition geformt werden.

- **Wäre eine Vorlage (Skizze, Foto, ...) sinnvoll?**

In vielen Fällen kann sich eine grobe Skizze oder ein Foto als sehr nützlich erweisen, da man dann von Beginn an gewisse Richtlinien hat und keine Probleme auf Grund von falschen Proportionen oder fehlenden Details bekommt.



- **Organisch oder industriell?**

Soll das Model organisch und natürlich wirken (Person, Tier, Alien, Pflanze) so kommt es nicht so sehr auf die 100%ig korrekte Position der einzelnen Vertices an. Modelliert man hingegen industriell gefertigte / künstliche Objekte (Möbel, Auto, Raumschiff, Gebäude) so sind exakte senkrechte und parallele Linien sehr von Bedeutung.

- **Einsparungen von Polygonen durch Texturen möglich?**

Modelliert man im Low Poly Bereich, so muss man versuchen, so viele Polygone wie möglich einzusparen. Dies lässt sich oftmals, besonders bei kleinen Details, durch geschickte Texturierung bewerkstelligen. Eine Gürtelschnalle, Schnürsenkel oder ein Riss in der Wand beispielsweise lassen sich sehr gut durch eine Textur darstellen.

Im nachfolgenden Beispiel soll ein **Eisbär für ein Computerspiel** erstellt werden.

Die Antworten auf diese Fragen lauten somit:

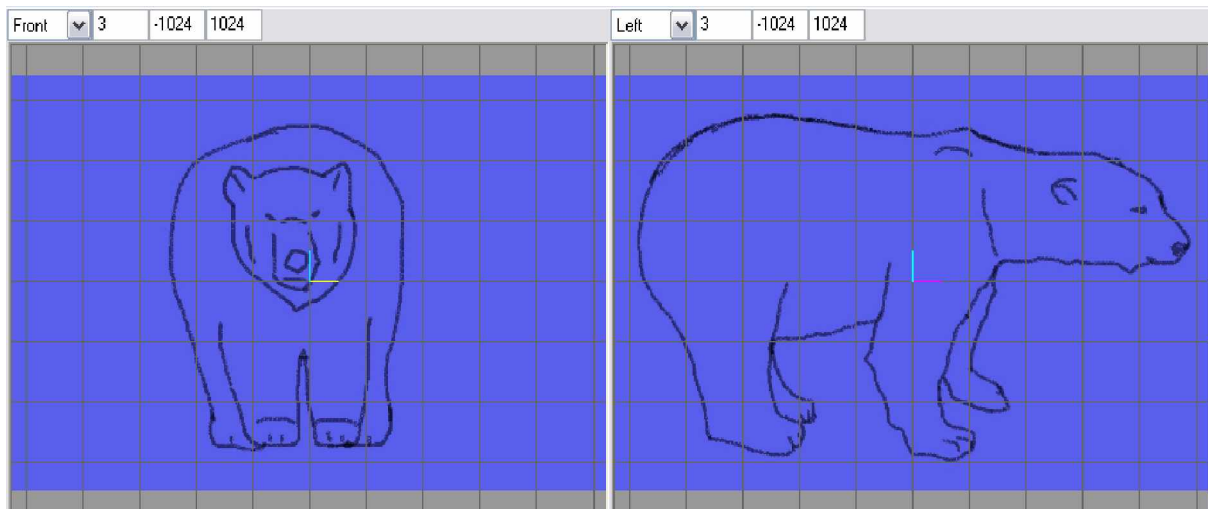
- Low Poly (maximal 300 Vertices), da für ein Computerspiel
- Zwar wird der Eisbär vorerst nur statisch benötigt, jedoch wäre es besser, die Option zum Animieren offen zu lassen und ihn in einer Standartpose zu erstellen
- 2 Skizzen, Profil- und Frontansicht
- Organisch, da es sich hierbei um ein Tier handelt
- Fell, Augen & Schnauze werden als Textur erstellt um Polygone zu sparen



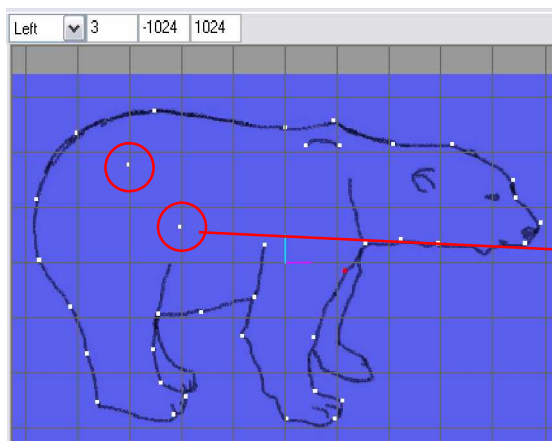
4.b Modelling

1. Bevor es mit dem eigentlichen Modelling losgeht, werden zwei grobe Skizzen eines Eisbären im Profil und von vorne erstellt, an denen man sich später orientieren kann. Dies kann in jedem beliebigen 2D-Grafikprogramm erfolgen. Wichtig ist, dass die Bilder anschließend in einem gängigen Grafikformat wie **JPG/BMP/PNG/etc.** gespeichert werden.

Diese Bilder werden nun in MilkShape über **Rechtsklick auf das entsprechende Fenster** → „Chose Background Image...“ importiert.



2. Nun müssen in Handarbeit an markanten Stellen Vertices verteilt werden, die sich an den Skizzen orientieren sollten. Dabei ist es nicht sonderlich schlimm, wenn das Model Kanten hat. Diese sind im Endeffekt unvermeidbar, sodass man nur versuchen kann diese auf



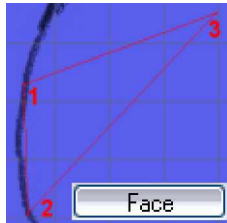
ein erträgliches Maß zu reduzieren, wobei das gesetzte Ziel von 300 Vertices nicht überschritten werden sollte.

Es sind einige zusätzliche Punkte notwendig, die nicht auf der Skizze ersichtlich sind um vollständige, abgeschlossene Körperteile zu erhalten.

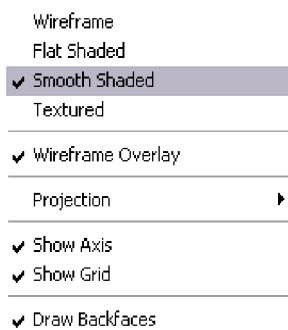
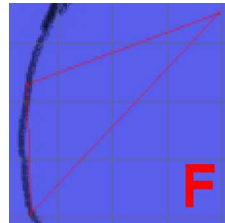


3. Als nächstes müssen aus diesen Punkten Flächen werden. Dies kann über zwei Möglichkeiten erfolgen:

1. Über das „**Face**“-Werkzeug jeweils drei Vertices hintereinander, gegen den Uhrzeigersinn anklicken.

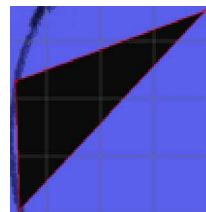


2. Durch selektieren von 3 Vertices mit dem „**Select**“-Werkzeug und anschließendem Drücken der Taste **F**

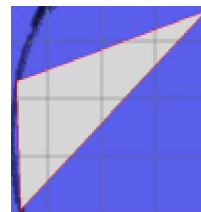


Die Vorgehensweise ist Geschmackssache und kann von je nach Gefallen gehandhabt werden. Empfehlenswert ist das Wechseln in den „**Smooth Shaded**“ Modus, sowie die Aktivierung von „**Draw Backfaces**“ um die Ausrichtung der Vertices sehen zu können.

Wurde versehentlich eine zugewandte Rückseite erstellt, lässt sich diese im noch selektierten Zustand problemlos Durch drücken von **Strg+Shift+F** (“Reverse Vertex Order“) umkehren

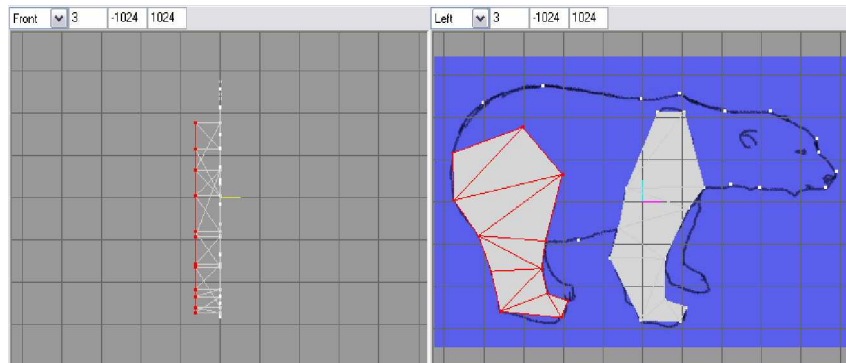


è **Strg+Shift+F** è

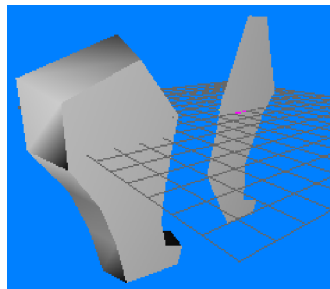


Auf diese Weise werden nun die Gliedmaßen geformt. Nach diesem Durchgang sind diese jedoch noch absolut flach, da alle Punkte in nur einer Ebene liegen.

4. Aus diesem Grund kommt nun das „Extrude“ Werkzeug zum Zuge. Dazu werden zuerst alle Vertices des Hinterbeines selektiert und dann (mit aktivierten Extrude-Button) in der Front-

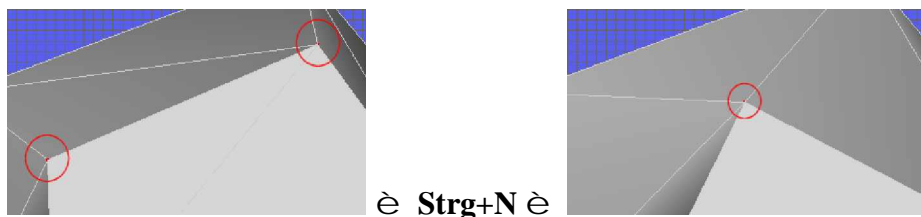


Ansicht etwas nach außen gezogen und anschließend über „Scale“ etwas verkleinert

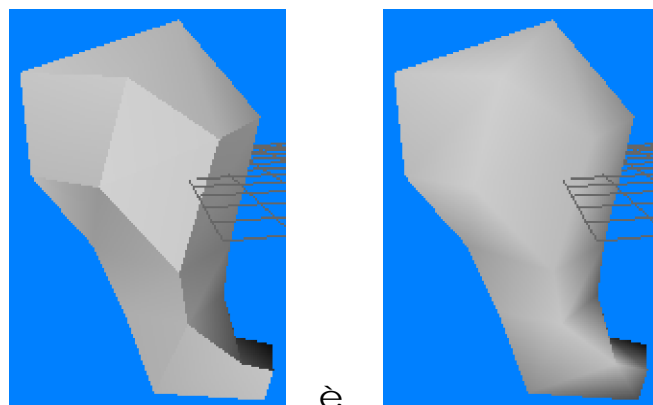


Der aktuelle Stand im 3D-Fenster betrachtet

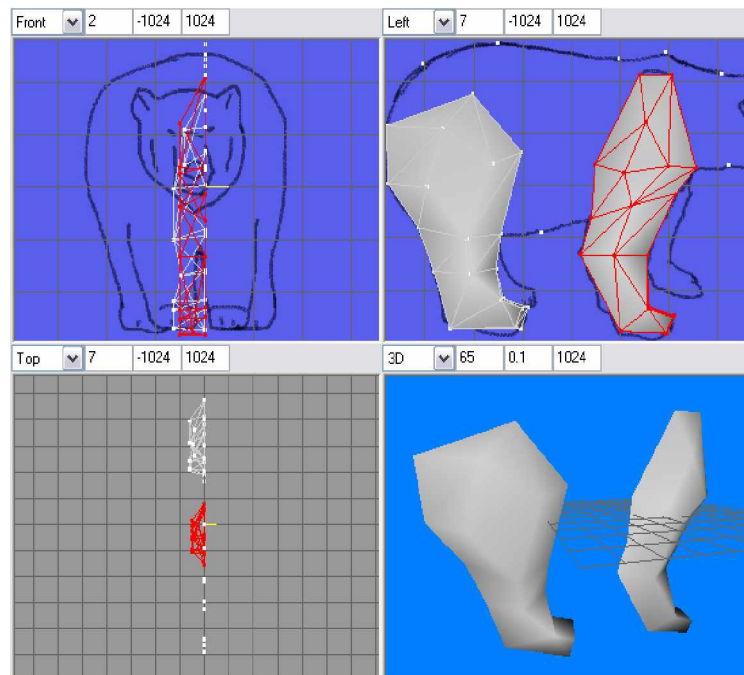
Da die extrudierte Fläche deckgleich mit dem Ursprung ist, wirkt dies sehr künstlich und eher roboterhaft. Deshalb muss sie noch bearbeitet werden, indem man einige der Vertices zusammenfasst und verschiebt.



Dass die Schattierung zu diesem Zeitpunkt noch absolut nicht stimmt, ist nicht wichtig; dies wird erst im Anschluss korrigiert. Zunächst die Vertices über das „**Groups**“-Menü zu einer Gruppe zusammengefasst und mit einen kennzeichnenden Namen (z.B. `bein_hr`) versehen. Über das gleiche Menü wird diese Gruppe zusätzlich zu einer Schattierungsgruppe hinzugefügt (Smoothing Groups → Assign → 1) und zum Schluss alle doppelten Vertices verschmolzen (**Strg+W**)

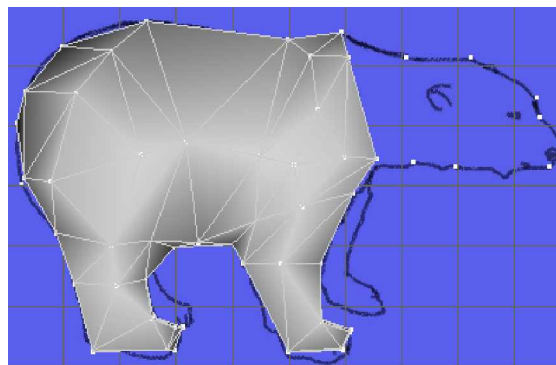
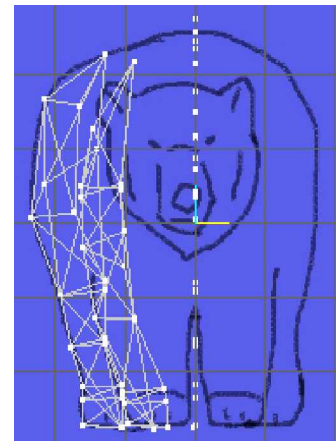


5. Das gleiche wird nun mit dem **Vorderbein** gemacht, sodass als Zwischenstand folgendes vorhanden ist:



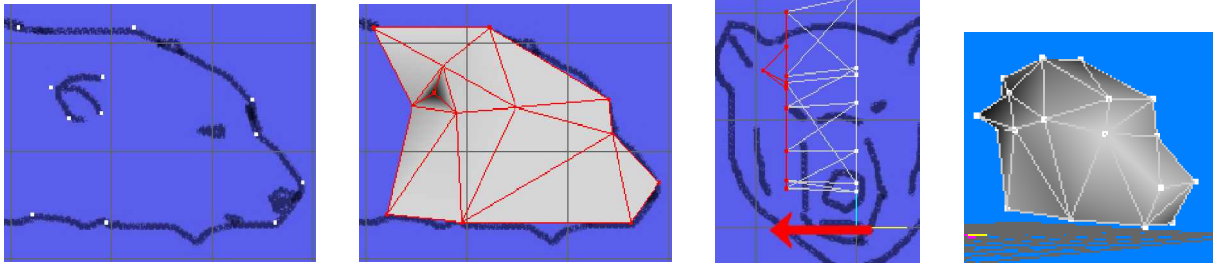
Im nächsten Schritt werden die Beine in der Frontansicht in die richtige Position verschoben, sodass sie in etwa zur Skizze passen.

6. Mit dem **Körper** wird nun auf gleiche Weise verfahren, wie mit den Beinen: Vertices zu Faces verbinden, über die Front- und Top-Ansicht nach außen verschieben. Stellt man fest, es fehlt an der einen oder anderen Stelle noch ein Vertex, da sich sonst keine Rundungen nach außen erstellen lassen, so kann dieser ohne Probleme hinzugesetzt werden.

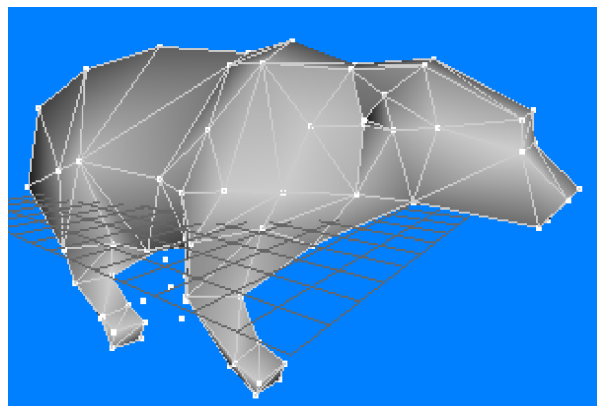




7. Anschließend mit dem **Kopf**: Vertices setzen, verbinden, extrudieren, nachkorrigieren,

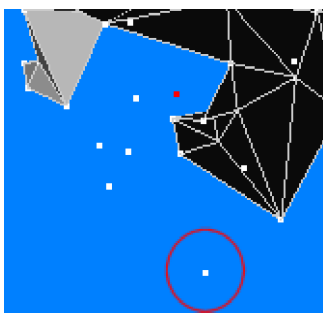


Gruppe erstellen, benennen, zu Smoothing Group 2 hinzufügen & welden (Strg+W), mit dem Körper verbinden.

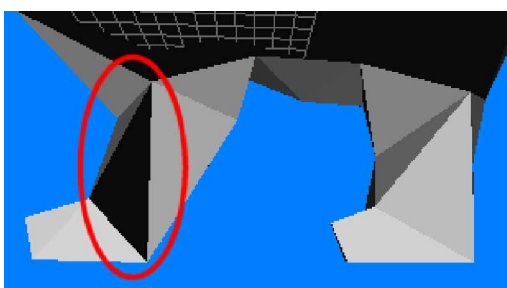


Dass bis jetzt alles als Hälfte entstand, war Absicht. Denn es ist viel einfacher und besser, spiegelbare Objekte zur Hälfte zu erstellen, duplizieren, spiegeln Schnittstelle verbinden und Feinänderungen vorzunehmen. Was zu diesem Schritt noch fehlt, sind vollständige Gliedmaßen und werden deshalb nun vervollständigt.

8. Zwei Tipps:



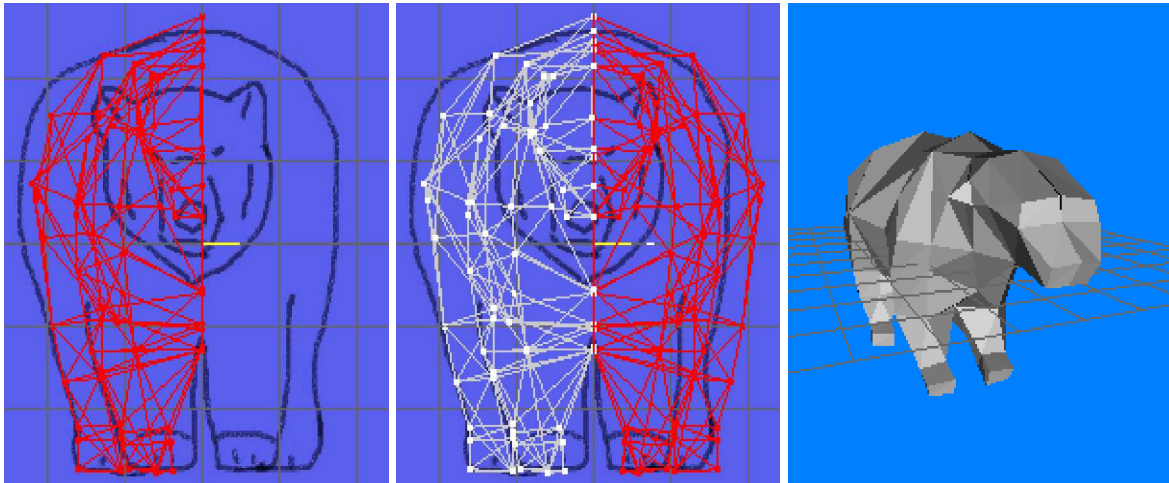
1. Fallen **übrig gebliebene Vertices** auf, so lassen sich diese durch einen Trick leicht entfernen. Dazu selektiert man alle Faces (Select→Face→Rahmen um alles ziehen) versteckt nun diese Auswahl (Strg+H) und selektiert und entfernt nun die übrig gebliebenen Vertices. Jetzt nur noch das versteckte sichtbar machen (Strg+Shift+H)



2. Es ist sinnvoll ab und an in den „**Flat Shaded**“-Modus zu wechseln, da man so verdrehte Faces besser erkennen kann. Diese dann einfach selektieren und über Strg+Shift+F wenden.

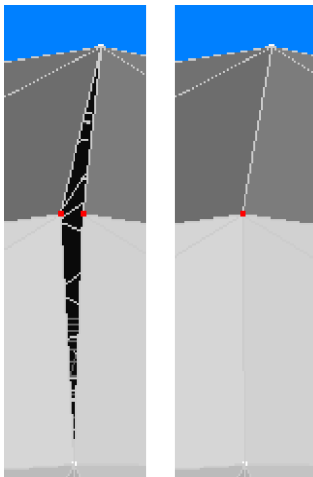


9. Sobald der halbe Eisbär fertig ist, kann gespiegelt werden.



Alles **selektieren**, **duplizieren** (Strg+D), **spiegeln** (Menü: Vertex → Mirror Left <-->Right)

Dort wo die Vertices nicht exakt auf der x-Achse lagen, weil man unvorsichtig verschoben hat, ergeben sich nicht saubere Kanten.



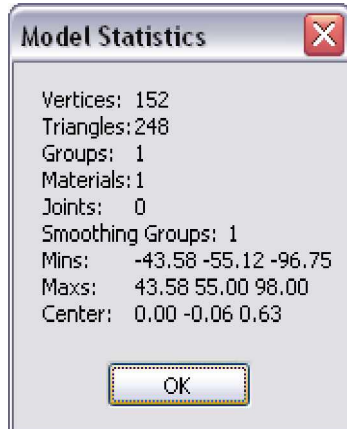
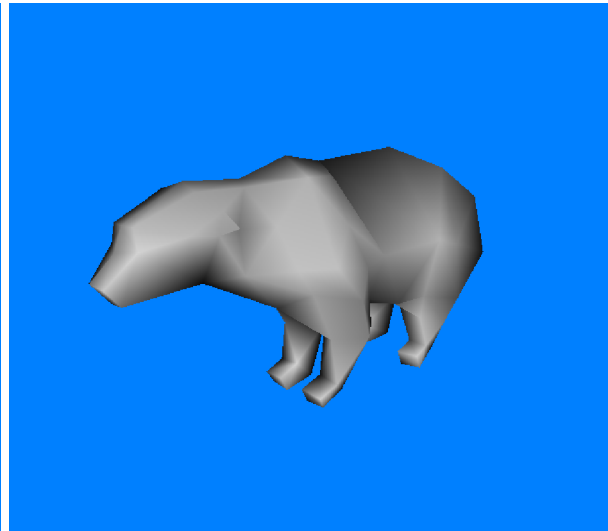
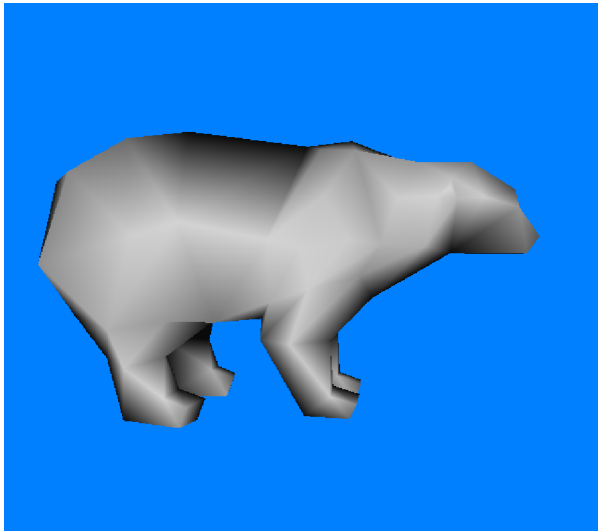
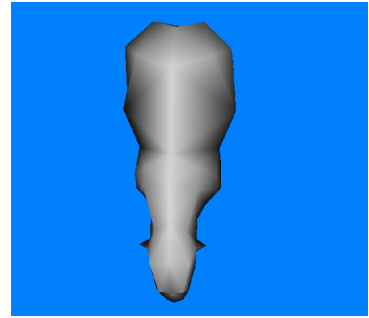
In diesem Fall am Kopf:

Dies ist jedoch nicht schlimm, da man diese Vertices einfach selektieren und über einander schieben kann (Strg+N).

An einigen Stellen kann es sinnvoll sein von Hand nachzumodellieren, da kein Lebewesen exakt spiegelgleich ist. So wäre zusammenfassen, erweitern oder verschieben, je nach ästhetischen Aspekten zu empfehlen.



10. Zum Schluss alles **selektieren** (Strg+A) zu einer **Smoothing Group** hinzufügen, da es sich nun um nur noch ein einziges Objekt handelt, das keine harten Schattengrenzen erfordert, (Smoothing Groups → Assign → 1) und **welden** (Strg+W).



Das Ergebnis kann sich schon sehen lassen.

Die **Model Statistics** lassen sich über „Menü: Tools → Model Statistics“ aufrufen.

Unter anderem findet man hier „Triangles“ (=Dreiecke).

Die Vorgabe von maximal 300 Faces wurde also eingehalten.



4.c Skinning

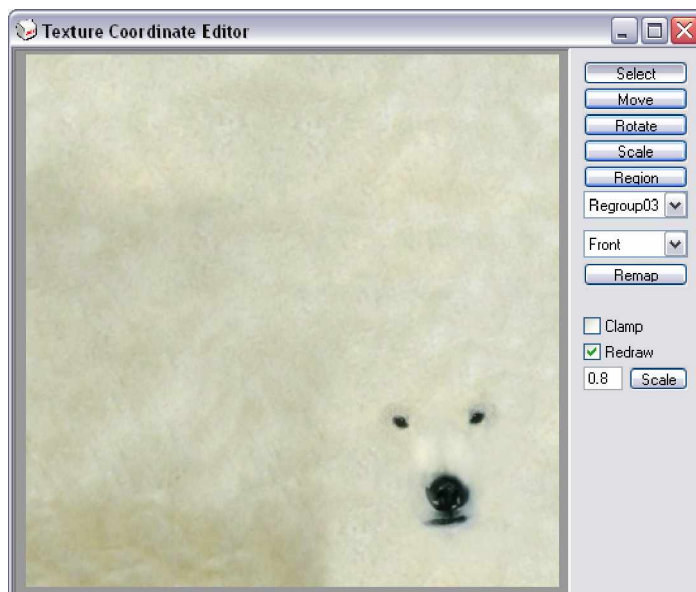
Als letzter Schritt wird dem Modell eine Textur verpasst, die den Bären mit Fell, Augen und Schnauze ausstatten wird.

Die Textur muss eine Grafikdatei, die als Höhe und Breite eine 2er-Potenz (2^x) hat, da es ansonsten zu Darstellungsfehlern kommen kann. (Bsp.: 128x128, 256x64, 512x512, ...)

Normalerweise ist es üblich, die Textur so zu erstellen, dass man jede Gruppe (Körperteile, Kleidung, Zubehör, etc.) einzeln in der Grafikdatei abbildet und diese dann separat darauf verteilt. In diesem Fall ist es jedoch deutlich ratsamer die Textur als nahtlose Fellgrafik zu erstellen und einen Bereich für die Kopfgrafik zu belassen, da sich das Fell kantenlos über den gesamten Körper erstreckt.



Über **Materials** → **New** → **<none>** → **Grafikdatei** wird nun die neue Textur angelegt. Die unterschiedlichen Gruppen werden nun nicht mehr benötigt und können über **Strg+A** (alles selektieren), **Groups** → **Regroup** zu einer einzigen Gruppe zusammengefasst werden. Nun kann man die Textur dieser Gruppe über **Materials** → **Assign** zuweisen.

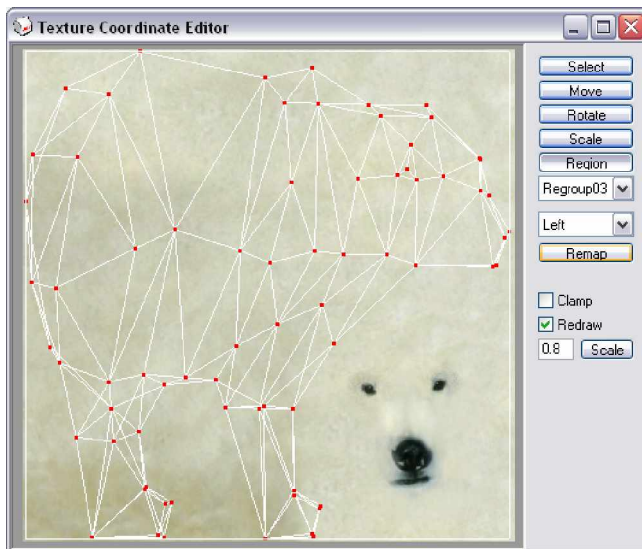


Jetzt muss diese Textur noch richtig auf dem Model verteilt werden. Dies geschieht im „**Texture Coordinate Editor**“ (Strg+T)

Hier wird die Sichtperspektive „**Left**“ gewählt, da die Textur von der Seite auf das Model übertragen wird.

Anschließend zieht man mit dem „**Region**“-Werkzeug einen Rahmen um den Bereich, den man für die aktuelle Gruppe verwenden möchte.

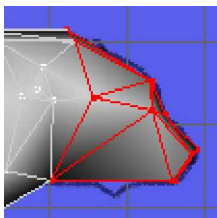
Da es nun nur noch eine Gruppe gibt, wird der Rahmen um das gesamte sichtbare Bild gezogen.



Ein Klick auf „**Remap**“ projiziert nun das Gittermodell auf die Grafik. Es ist nicht sonderlich wichtig, dass dabei die Proportionen nicht stimmen. Im Endeffekt wird die Grafik über das Model gestreckt, sodass dies beim Fell nicht auffallen wird.

Ist bei „**Redraw**“ ein Haken gesetzt kann man die Änderungen in Echtzeit im 3D-Fenster mitverfolgen (vorausgesetzt dort

ist der Darstellungsmodus „**Textured**“ eingestellt).

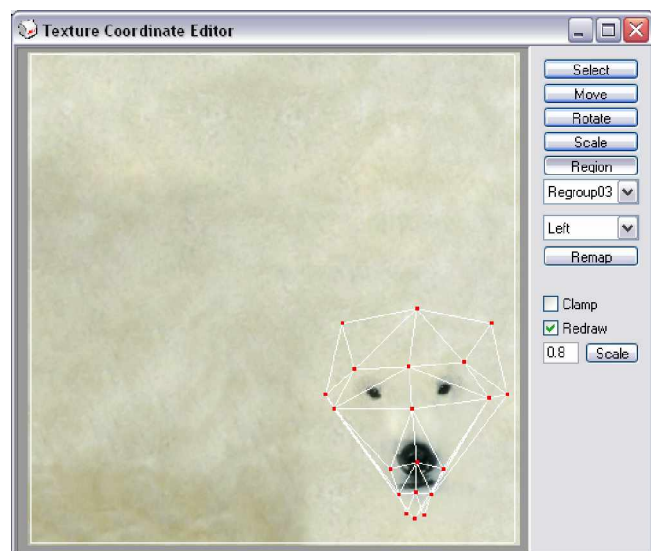


Zum Schluss müssen nur noch die Augen und die Schnauze auf die richtige Position gebracht werden. Dazu

selektiert man in einem der 2D-Fenster den Kopfbereich, sodass im Textureditor nur noch dessen Texturpunkte zu finden sind. Nun definiert man einen neuen Bereich über

„**Region**“ um die rechte untere Ecke herum, wählt „**Front**“ und klickt auf „**Remap**“.

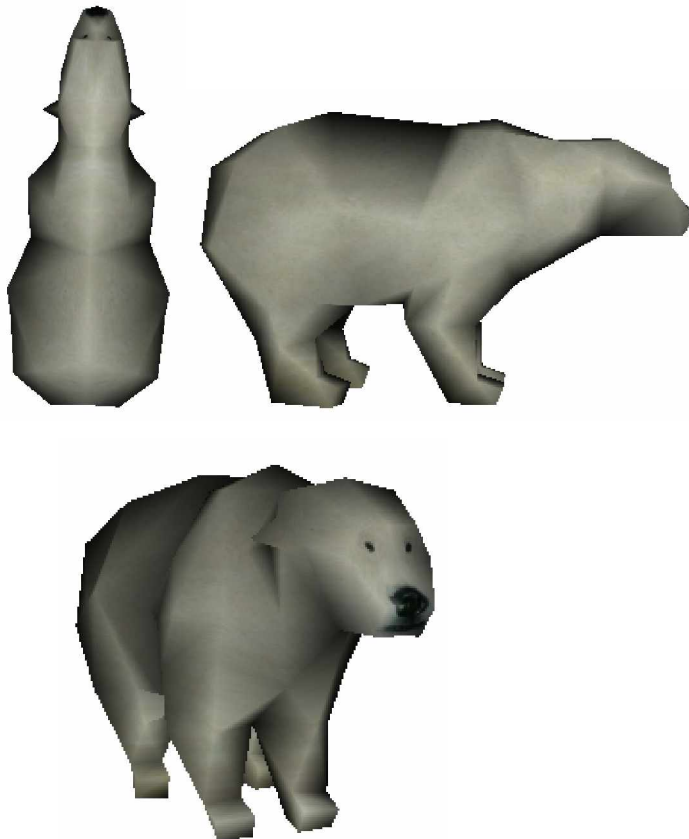
Die automatische Projektion ist selten ideal, sodass man die Punkte nur noch nach Gefallen zu Recht schieben muss, was durch die Echtzeitvorschau im 3D-Fenster kein Problem darstellen sollte.



Ein letzter prüfender Blick

...

und das Model ist fertig gestellt!



... und befindet sich bereits wenig später im Spiel.





Quellen

**Sämtliche Informationen stammen aus dem eigenen Wissen.
In seltenen Fällen wurde die „MilkShape3D Helpfile“ zur Hilfe genommen.**

<http://www.swissquake.ch/chumbalum-soft/files/download.php?file=MS3DHelp.chm>

von

<http://www.MilkShape3d.com/>

Bildquellen

Selbst angefertigte Screenshots aus der Software **MilkShape3D 1.7.2**

Nähere Informationen unter <http://www.MilkShape3d.com/>

2 Bilder aus der Google Bildersuche zur Untermalung

<http://www.google.de/imghp>