Blender Animationen für XNA - Tutorial

Andrea Fuchsloch, Katja Weidner {fuchsloch|weidner}@fzi.de

Zentrum für Softwarekonzepte (ZfS) Karlsruhe

Abstract

Dreidimensionale Modelle werden in der Regel in einem 3D-Modellierungs-Tool wie bspw. Blender erstellt. Die Entwicklung von Spielen andererseits erfolgt in einer Entwicklungsumgebung wie dem XNA Game Studio. Die Integration der beiden Anwendungen stellt dabei eine Herausforderung für den Entwickler dar. Dieses Tutorial beschreibt, wie Animationen dreidimensionaler Modelle in Blender erstellt werden müssen, um diese in XNA verwenden zu können. Des Weiteren wird darauf eingegangen, wie diese Animationen innerhalb des XNA-Frameworks angesprochen und abgespielt werden können.

Animationen ermöglichen es, nahezu jedes Element einer Szene „zum Leben zu erwecken“. Eine Animation umfasst dabei eine visuelle Veränderung eines Objektes von Frame zu Frame. Auf diese Weise wird nicht nur eine hohe Komplexität innerhalb von Spielen erreicht, sondern auch ein beträchtliches Maß an Realismus in der Darstellung gewonnen. (1)

Das XNA Framework bietet standardmäßig nur eine eingeschränkte Unterstützung von Animationen. Das Framework bietet zwar ein Objekt `Model`, das es erlaubt, Animationsdaten aus den Formaten Autodesk FBX Format (.fbx) (2) und dem DirectX File Format (.x) (3) in der Content Pipeline zu speichern. Es beinhaltet aber keine Klassen, um diese Animationen zur Laufzeit abspielen zu können. (4)

Das vorliegende Tutorial bezieht sich auf das 3D-Modellierungs-Tool Blender in der Version 2.45 (5) sowie das XNA Game Studio 2.0 (6). Anhand des Modells eines Schmetterlings wird in einem durchgehenden Beispiel eine „Flug“-Animation für eben diesen Schmetterling in Blender erstellt. Dabei wird vor allem auf die Besonderheiten eingegangen, die bei der Erstellung von Animationen in Blender beachtet werden müssen, um die Animationen später innerhalb des XNA-Frameworks ansprechen und abspielen zu können. Das verwendete Schmetterlings-Modell kann von der Webseite des Zentrums für Softwarekonzepte (ZfS) Karlsruhe heruntergeladen werden (7).

1 Grundlagen

Es gibt verschiedene Möglichkeiten Animationen zu erstellen. Die Art der gewählten Animationstechnik wird dabei in erster Linie vom gewünschten Einsatz der Animation bestimmt.

Um eine große Anzahl an gleichen Objekten zu animieren, bei denen es nicht auf das Verhalten der einzelnen Objekte, sondern auf das Verhalten der Gesamtheit ankommt, bietet es sich an, ein Partikelsystem einzusetzen. Partikelsysteme werden daher häufig verwendet um Schnee, Regen oder Rauch darzustellen.

Für die Animation von dreidimensionalen Modellen sind besonders die Techniken der Schlüsselbildanimation (Keyframe Animation) und die Skelett-basierte Animation von Bedeutung, die meist in Kombination eingesetzt werden. Im Folgenden wird auf diese beiden Techniken detailierter eingegangen.

1.1 Keyframe Animation

Die Schlüsselbildanimation stammt ursprünglich aus der Zeichentrickproduktion. Dabei wird mit Hilfe der sogenannten Schlüsselbilder ein grober Bewegungsablauf vorgegeben, ohne dass bereits die dazwischenliegenden Einzelbilder gezeichnet wurden.

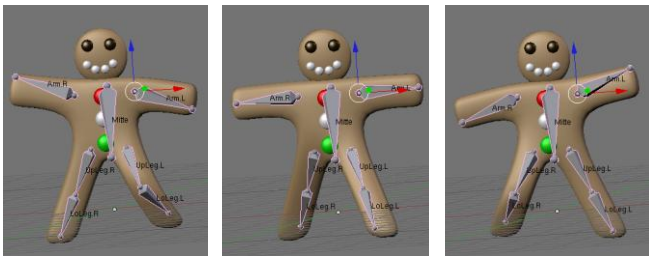
Frame 1 Frame 21 Frame 41

Abbildung 1: Schlüsselbilder eines Zeichentrickfilms. (1)

Im Kontext dreidimensionaler Modelle ist der Begriff Schlüsselbild oder auch Keyframe etwas weiter zu fassen. Ein Keyframe speichert Parameter wie die Position, die Rotation und die Skalierung eines Objektes. Für die Frames zwischen zwei Keyframes werden die Werte dieser Parameter durch Interpolationsmethoden berechnet. Auf diese Weise entsteht ein flüssiger Bewegungsablauf. Des Weiteren können für die Werte der unterschiedlichen Parameter auch unterschiedliche Keyframes gewählt werden.

1.2 Skelett-basierte Animation

Bei der Skelett-basierten Animation werden die dreidimensionalen Modelle um ein Skelett ergänzt. Die Idee der in die 3D-Modelle eingefügten Knochen lehnt sich an die Knochen der Biologie an.



Frame 1 Frame 21 Frame 41

Abbildung 2: Animation des Modells mit Hilfe des Skeletts. (8)

Das Skelett legt die Bewegungen zwischen den beweglichen Gliedern des Modells fest. Für jedes Gelenk im Skelett werden Parameter wie die Rotation, die Skalierung und die Translation in einer Matrix gespeichert. Die Animation wird erstellt, in dem das Skelett entsprechend positioniert wird. Das Modell verformt sich dann gemäß seinem Skelett. Die zwischen zwei Posen des Skeletts liegenden Frames werden durch die Interpolation der einzelnen Skelettposen berechnet. (9)

Im Rahmen der Skelett-basierten Animation wird zwischen Vorwärts-Kinematik und inverser Kinematik unterschieden. Dabei wird davon ausgegangen, dass die Knochen eines Skelettes zueinander in Beziehung bzw. in einer Hierarchie stehen. Wird bei einer Animation eine Bewegung eines Knochens festgelegt, wird diese auf die Untergruppen übertragen.

Vorwärts-Kinematik oder auch Forward Kinematics bedeutet, dass die Bewegung eines Knochens an seine Kind-Knochen weitergegeben wird. Wird beispielsweise der Oberarmknochen nach vorne beweget, werden sowohl der Unterarmknochen als auch der Handknochen ebenfalls nach vorne bewegt. (10)

Bei der inversen Kinematik erfolgt die Abarbeitung in der umgekehrten Reihenfolge, d.h. vom Kind-Knochen hin zum Vater-Knochen. Wird beispielsweise die Hand bewegt, werden die Knochen des Unter- und Oberarms entsprechend angepasst. Die inverse Kinematik ist dem natürlichen Denken nachempfunden und bei der Erstellung von Animationen einfacher zu handhaben. (1)

2 Rigging – Ein Skelett hinzufügen

Nachdem das Modell des Schmetterlings in Blender modelliert und seine Textur erstellt wurde (siehe hierfür (11)), soll er nun mit Hilfe von Animation „zum Leben erweckt“ werden. Konkret wird der Schmetterling, der auch in Abbildung 3 zu sehen ist, eine Animation erhalten, die genau einen Flügelschlag beinhaltet. Wird diese in einer Schleife abgespielt, scheint der Schmetterling zu fliegen.

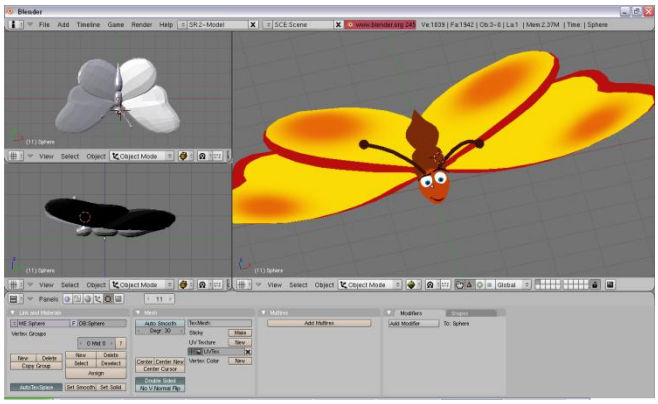


Abbildung 3: Ausgangssituation, - das Modell und seine Textur wurden erstellt.

Um eine Animation erstellen zu können, wird dem Modell zunächst ein Skelett, eine sogenannte Armature hinzugefügt, die wiederum aus mehreren miteinander verbundenen Knochen, den sogenannten Bones besteht. Das Ausrüsten eines Modells mit einer Armature wird Rigging genannt. (8)

Es bietet sich an, die Armature des Modells in Analogie zum Skelett im biologischen Sinn zu modellieren. Es ist allerdings zu bedenken, dass für die Animation eines 3D-Modelles häufig mehr Bones notwendig sind, als ursprünglich gedacht. Soll beispielsweise die Klei-

dung eines Modells unabhängig von den Gliedmaßen eines Modelles bewegt werden, müssen auch für diese entsprechende Bones modelliert werden. (10)

Um eine Armature hinzuzufügen, bietet es sich an, das Model in die „Wireframe“ Ansicht zu bringen. Der Cursor sollte an der Stelle platziert werden, an der der erste Bone seine Wuzel (root) haben sollte. Da es sich bei Blender um eine 3D Anwendung handelt, muss der Cursor in mindestens zwei Ansichten platziert werden. Wird der Cursor beispielsweise in der „View“ → „Top“ Ansicht platziert, sollte anschließend in der „View“ → „Side“ Ansicht sichergestellt werden, dass sich der Cursor an der gewünschten Stelle befindet.

Über „Space“ → „Add“ → „Armature“ wird der erste Knochen hinzugefügt. Wird im „Edit Mode“ die Spitze des Knochens ausgewählt, erscheint diese in Gelb und kann mit „G“ für „Grab“ bewegt werden. Mit der linken Maustaste kann die Spitze (Tip) des Knochens an der gewünschten Stelle platziert werden. (8) Der erste Knochen des Schmetterlings sollte wie in Abbildung 4 dargestellt, positioniert werden.

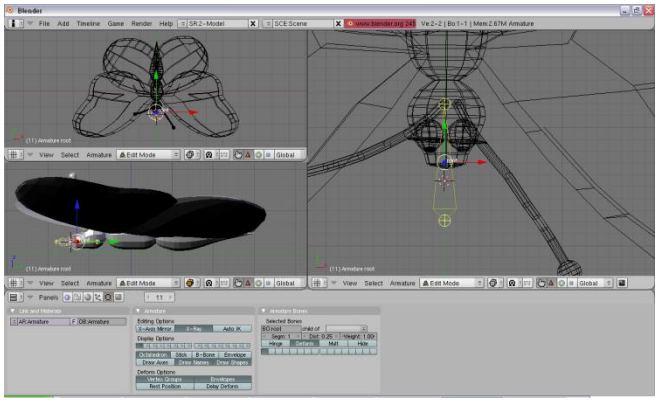


Abbildung 4: Position des Wurzelknochens des Schmetterlings.

Unter „Buttons Window“ im Panel „Armature“ sollte der „X-Ray“ Button aktiviert werden. Auf diese Weise ist die Armature immer sichtbar, auch wenn das Modell nicht im „Wireframe Mode“ dargestellt wird. Des Weiteren sollten die Bones sprechende Namen erhalten. Die Namen der Bones können im „Buttons Window“ im Panel „Armature“ mit Hilfe des Buttons „Draw Names“ angezeigt werden. Um den Namen eines Bones zu ändern, muss der entsprechende Knochen ausgewählt werden. Im Panel „Armature Bones“ kann unter „Selected Bones“ im Feld „BO:“ ein sprechender Name für den Knochen eingegeben werden. Abbildung 5 zeigt die beschriebenen Einstellungen.

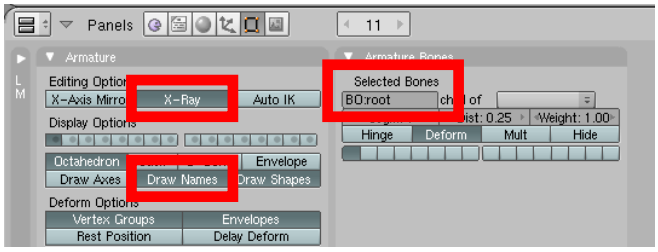


Abbildung 5: Einstellungen bei der Modellierung des Skelettes.

Für die Verwendung der Animation im XNA Framework ist es sehr wichtig, dass ein zusammenhängendes Skelett modelliert wird. Keinesfalls darf das Skelett Bones enthalten, die nicht mit anderen Bones der Armature verbunden sind. Innerhalb von Blender ist die Erstellung einer solchen, nicht zusammenhängenden Armature möglich und stellt auch bei der Erstellung einer Animation innerhalb von Blender kein Hindernis dar. Das XNA Framework dagegen gibt folgende Fehlermeldung aus: „Node has more than one BoneContent child. Unable to determine which one is the skeleton root. “

Um eine zusammenhängende Armature zu modellieren, sollten weitere Bones immer an die Spitze bereits hinzugefügter Bones angehängt werden. Auf diese Weise stehen die Bones in einer Beziehung zueinander. Der neu angehängte Knochen wird zum Kind-Knochen des bestehenden Bones. Wird im „Edit Mode“ die Spitze des bestehenden Knochens ausgewählt, kann durch drücken der „E“ –Taste ein weiterer Knochen, der child-bone, aus dem bestehenden Knochen heraus extrudiert werden. Abbildung 6 zeigt, wie der aus dem „root“ Knochen des Schmetterlings die Knochen „Hals“, „Brust“ und „Schwanz“ extrudiert wurden.

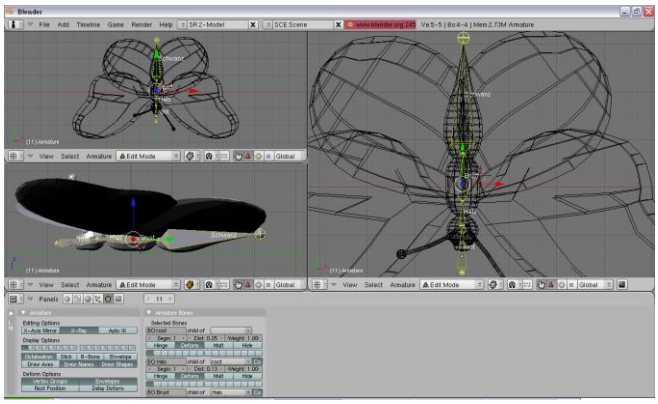


Abbildung 6: Hinzufügen weiterer Bones durch Extrudieren.

Um zu prüfen, ob eine zusammenhängende Armature modelliert wurde, bietet es sich an, in den „Pose Mode“ der Armature zu wechseln. Der Pose Modus ist ein spezieller Bearbeitungsmodus, den nur Armatures besitzen (8). Nun können die einzelnen Knochen mit „G“ für „Grab“ etwas bewegt werden. Die verbundenen

Bones sollten sich dabei ebenfalls mit bewegen. Da zu diesem Zeitpunkt die Position der Bones noch nicht verändert werden soll, sollte diese Bewegung wieder rückgängig gemacht werden.

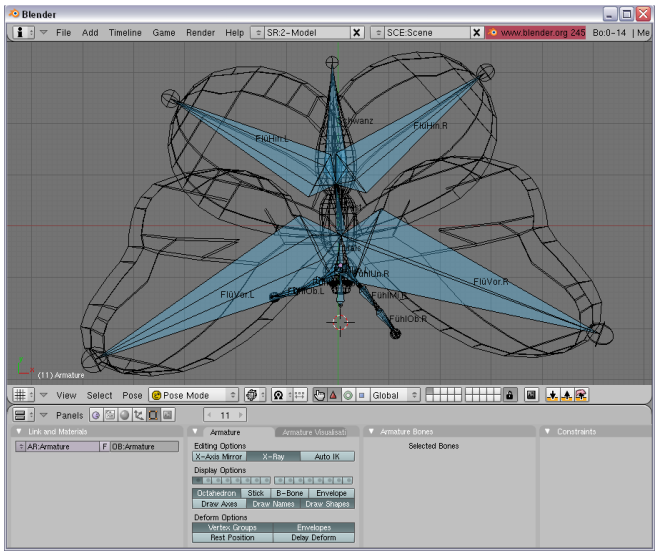


Abbildung 7: Modell des Schmetterlings einschließlich seiner Armature.

Nachdem das Skelett des Schmetterlings vollständig erstellt wurde, müssen die Bones orientiert werden, d. h. sie müssen gegenüber dem Modell in ihrer Achse rotiert werden. Wurde die Armature in den „Edit Mode“ versetzt, können mit Hilfe von „A“-„A“ alle Bones ausgewählt werden. Anschließend sollte die Funktion „Armature“ → „Clear Bone Roll Angle“ angewendet werden. Detaillierte Anleitungen für das Erstellen einer Armature sind unter (10) und (8) zu finden.

3 Skinning - Skelett und Modell verbinden

Bislang bewirkt eine Bewegung der Bones noch keine Verformung des Modells. Um dies zu erreichen, muss die Armature mit dem Modell verbunden werden, was auch Skinning genannt wird. Bei diesem Vorgang werden die Vertices zu den Bones zugewiesen, so dass sich die Vertices mit dem Knochen mit bewegen. Die Armature wird zum „Modifier“ des Modells.

Um diese Verbindung herzustellen, sollte zunächst das Modell in die „Wireframe“ Ansicht versetzt und ausgewählt werden. Anschließend muss mit „Shift“ und der rechten Maustaste die Armature ausgewählt werden. Das Modell sollte nun in der Farbe magenta und die Armature in pink erscheinen, wie dies in Abbildung 4 dargestellt ist. (8)

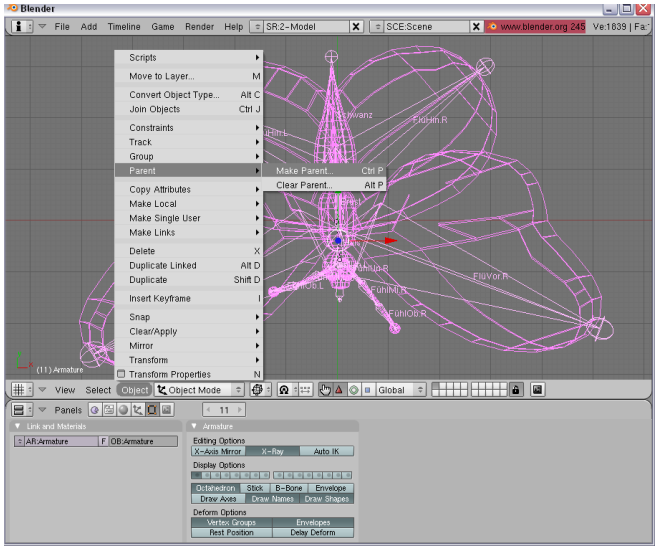


Abbildung 8: Die Armature wird zum Modifier des Modells.

Mit Hilfe der Funktion unter „Object“ → „Parent“ → „Make Parent“ wird die Armature zum Elternteil des Modells ernannt (8). Die Dialogbox „Create Vertex Groups?“ bietet unter anderem den Eintrag „Create From Closest Bones“. Diese Funktion ordnet bereits Vertices zu Bones zu und bietet somit einen guten Ausgangspunkt für den nächsten Schritt.

Der Armature Parent muss nun zu einem echten Modifier des Modells werden. Hierfür muss das Modell ausgewählt werden und im Panel „Modifier“ neben der Angabe „Armature Parent Deform“ der Button „Make Real“ gewählt werden. Keinesfalls darf nun der Button „apply“ angewendet werden. Für weitere Informationen zum Thema „Modifiers“ siehe (12).

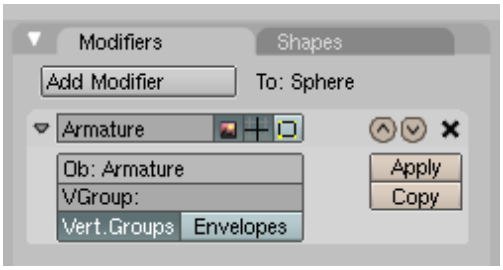


Abbildung 9: Die Armature wird zum Modifier des Modells.

Der Button „Envelopes“ ist standardmäßig ausgewählt. Der Button sollte deaktiviert werden (siehe Abbildung 9), da im Folgenden dieser Mechanismus nicht verwendet wird.

4 Weight Painting

Im nächsten Schritt werden die bereits automatisch erstellten Vertexgruppen nachgearbeitet. Mit Hilfe der Vertexgruppen wird der Einfluss festgelegt, den ein bestimmter Bone auf einen bestimmten Vertex ausübt. Blender bietet hierfür eine intuitive Technik, das sogenannte Weight Painting, an (8). Dabei wird für jeden

Bone mit Hilfe eines Pinsels die Einflussstärke auf das Modell aufgemalt. Die Farbe Rot steht für einen starken Einfluss des Bones auf den Vertex, die Farbe Blau hingegen für keinen Einfluss des Bones auf den Vertex. Die entsprechenden Vertexgruppen werden automatisch von Blender erstellt.

Um den Einfluss aufmalen zu können, muss die Armature ausgewählt und in den „Pose Mode“ versetzt werden (siehe Abbildung 10). Um in den „Weight Paint Mode“ zu gelangen, muss nun das Modell ausgewählt werden und ebendieser Modus gewählt werden (siehe Abbildung 11). Nun kann ein Bone ausgewählt werden um für eben diesen Knochen die Vertices zu bemalen, die mit dem Knochen bewegt werden sollen.

Abbildung 10: Armature im "Pose Mode".

Abbildung 11: Das Modell im Modus „Weight Paint“.

Wie bereits erwähnt, wird die relative Einflussstärke des ausgewählten Bones auf einen Vertex durch die Farbe des Vertex symbolisiert. Blau bedeutet dabei keinen Einfluss (0%), Grün bedeutet 50% Einfluss und Rot 100% Einfluss. Vertices, die nur von einem Bone beeinflusst werden, werden dabei immer vollständig bewegt, egal ob der Wert des Gewichts "1" oder "0.001" beträgt. Erst wenn ein zweiter Bone den Vertex beeinflusst, werden die Gewichte miteinander verrechnet. (8)

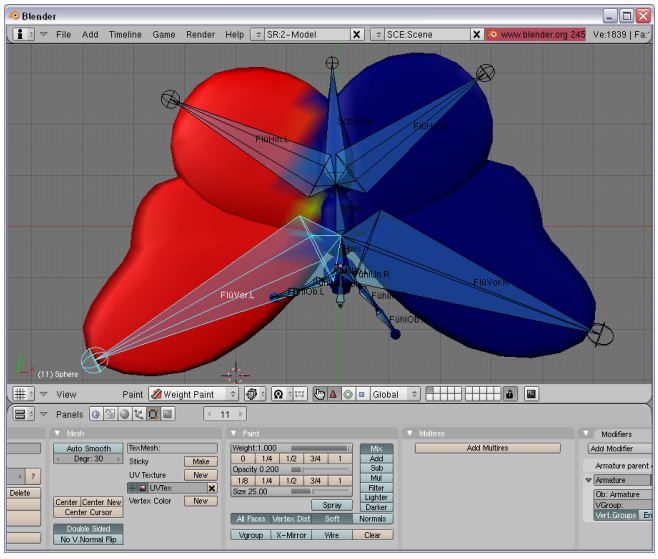


Abbildung 12: Weight Paint für den Knochen im Flügel vorne links des Schmetterlings.

Abbildung 12 zeigt das Modell des Schmetterlings im Weight Paint Modus. Für die Animation des Schmetterlings wurden, aus Gründen der Anschaulichkeit, nur die Gewichte eines und null verwendet. Die Abbildung zeigt den Einfluss, den der Knochen mit dem Namen „FlüVorne.L“ ausübt.

Der Einfluss wird mit Hilfe des Pinsels auf das Modell gemalt. Dieser lässt sich im „Buttons Window“ im Panel „Paint“ einstellen. Der Schieberegler „Weight“ gibt die Stärke des Einflusses an, der aufgemalt wird. Der „Opacity“ Regler die Deckkraft des Pinsels. Es bietet sich an, einen Bone nach dem anderen durchzugehen und die entsprechenden Bereiche zu bemalen. Dabei ist darauf zu achten, dass das Modell beim Malen für jeden Knochen von möglichst allen, aber mindestens von zwei Seiten bearbeitet wurde. (8)

Für die Zielplattform XNA gibt es beim Weight Painting in Blender eine Besonderheit zu beachten. XNA fordert, dass jeder Vertex zumindest einem Bone zugeordnet ist und für diesen ein Gewicht beinhaltet. Sollte dies nicht der Fall sein, wird vom XNA-Framework folgende Fehlermeldung ausgeben: „Error normalizing vertex bone weights. BoneWeightCollection does not contain any weighting values.“

Dieser Fehler kann mit Hilfe eines kleinen Tricks vermieden werden. Um sicher zu gehen, dass alle Vertex min. einem Bone mit einem definierten Gewicht zugeordnet sind, werden alle Vertex eines Modells einem bestimmten Knochen mit 100% Einfluss zugeordnet. Dieser Knochen sollte vorzugsweise gleichzeitig der Wurzelknochen des Skelettes sein. Wurde der Wurzelknochen ausgewählt und alle sichtbaren Vertices rot bemalt, besteht der Trick darin, den Wurzelknochen zu bewegen. Vertices, denen noch kein Gewicht zugeordnet wurde, machen die Bewegung nicht mit, sonder bleiben „stehen“. Auf diese Weise können auch schwer erreichbare Vertices leicht bemalt werden.

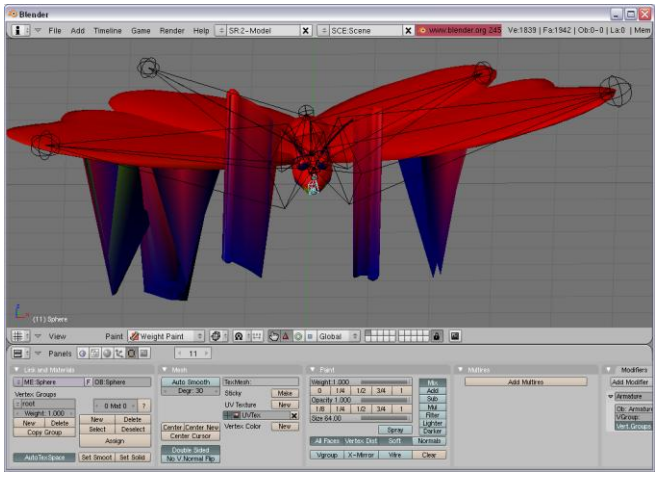


Abbildung 13: Vertices, für die noch kein Gewicht definiert wurde.

Der Knochen muss ausgewählt werden und kann nun mit „G“ bewegt werden. Der Knochen sollte nur auf

einer der drei Achsen gleichzeitig und nur um eine definiert Strecke bewegt werden. Dies kann erreicht werden, indem nach dem Grab ein „X“, „Y“ oder „Z“ gewählt und die Zahl „100“ eingegeben wird. Das Ergebnis sollte in etwa der Abbildung 13 entsprechen.

Die Vertices, die nicht dem Knochen zugewiesen wurden, haben ihre Position nicht geändert. In dieser Ansicht ist es sehr leicht, auch diese Vertices in der Farbe Rot zu bemalen. Sobald die Vertices ein Gewicht erhalten haben, ändern auch sie ihre Position. Anschließend sollte das Modell durch ein erneutes „graben“ wieder an die ursprüngliche Position gebracht werden. Wird diese Prozedur für jede der drei Achsen und jeweils für den positiven und auch den negativen Wertebereich durchgeführt, kann davon ausgegangen werden, dass allen Vertices ein Gewicht zugewiesen wurde.

Das Aufmalen des Einflusses eines Bones kann ein sehr aufwendiger Schritt sein, der unter Umständen wiederholt werden muss. Sollten sich beim anschließenden Posing bizarre Risse bilden, oder andere seltsame Verhaltensweisen zeigen, muss geprüft werden, welcher Knochen diese Vertices beeinflusst, bzw. beeinflussen sollte.

5 Posing

Inzwischen verfügt der Schmetterling über ein Skelett anhand dessen er sich verformen lässt. Der nächste Schritt besteht darin, das Modell in bestimmte Positionen zu setzen und entsprechende Keyframes einzufügen. Doch zunächst soll der Blender Workspace angepasst werden. Es bietet sich an, folgende vier „Window types“, wie in Abbildung 14 dargestellt, anzuordnen. Den „Action Editor“ links, die „3D View“ darunter, die „Timeline“ unter beiden und das „Buttons Window“ darunter.

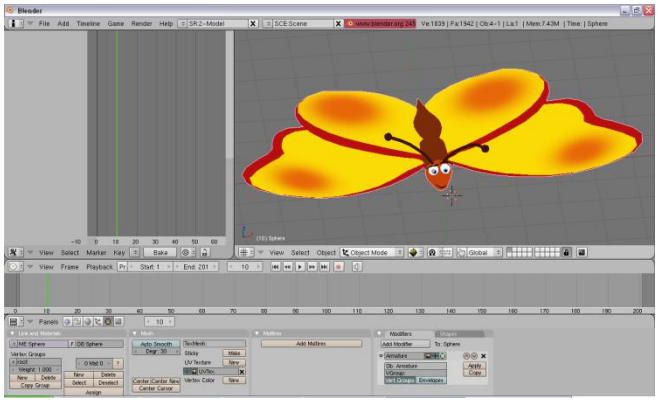


Abbildung 14: Workspace für das Positionieren der Armature.

In der „3D View“ kann für das Modell die Ansicht „Solid“ oder „Textured“ gewählt werden. Die Armature dagegen muss in den „Pose Mode“ gebracht werden.

Animationen, die sich später in XNA ansprechen und abspielen lassen, werden in Blender „Actions“ genannt. Der erste Schritt sollte daher darin bestehen, eine neue Aktion anzulegen, in dem im „Action Editor“ der Button mit den beiden Pfeilen (siehe Abbildung 15) ausgewählt wird. Die neue Aktion sollte einen sprechenden Namen wie beispielsweise „flying“ erhalten, da sie auch innerhalb des XNA Frameworks über diesen angesprochen wird.

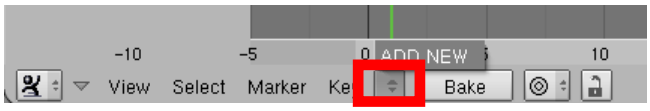


Abbildung 15: Eine neue Aktion hinzufügen.

Der Aktion sollten nun immer dann Keyframes hinzugefügt werden, wenn sich die Position, die Rotation oder Skalierung eines Bones ändern soll. Für die Animation des Schmetterlings werden drei Keyframes eingefügt, um einen Flügelschlag zu animieren. Die für eine flüssige Bewegung notwendigen Positionen zwischen den Keyframes werden von Blender automatisch erstellt.

Das Vorgehen bei der Erstellung von Keyframes sollte immer derselben Reihenfolge entsprechen. Als erstes wird die Nummer des Frames festgelegt. Wird die Nummer des gewünschten Frames direkt in das Feld für den aktuellen Frame der „Timeline“ eingegeben, springt Blender automatisch an diese Stelle. Abbildung 16 markiert das Feld des aktuellen Frames. Der erste Keyframe sollte immer im ersten Frame abgelegt werden.

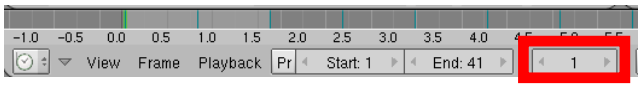


Abbildung 16: Aktueller Frame in Blender.

Der nächste Schritt besteht darin, die Knochen in die Ausgangslage der Bewegung zu rotieren. Im Fall des Schmetterlings wird die Bewegung beginnen, indem die Flügel erhoben sind. Die Armature muss sich hierfür im „Pose Mode“ befinden. Es bietet sich an, immer nur einen Bone zur gleichen Zeit mit „R“ zu rotieren.

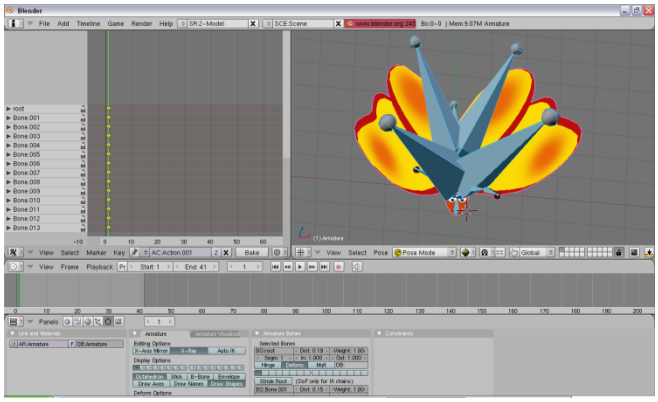
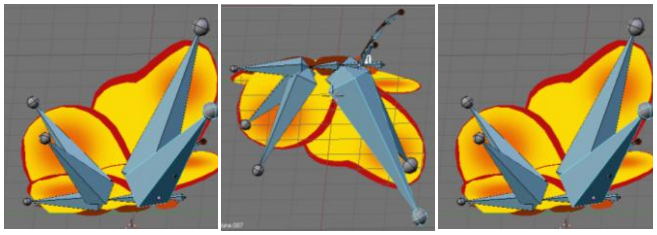


Abbildung 17: Ausgangsposition des Schmetterlings im ersten Keyframe.

Wurde der Schmetterling mit Hilfe der Knochen in die gewünschte Position gebracht, kann ein sogenannter IPO (Interpolation) Key eingefügt werden. Hierfür werden im „Pose Mode“ mit Hilfe der Taste „A“ bzw. „A“-„A“ alle Bones ausgewählt und mit der Taste „I“ für Insert Key ein Keyframe eingefügt. Es erscheint ein Dialogfeld um festzulegen, welche Parameter in dem neu zu erstellenden Keyframe gespeichert werden sollen. Da es sich um die Ausgangslage der Bewegung handelt, werden mit „LocRotScale“ die Position, die Rotation und die Skalierung aller Knochen gespeichert. Links im „Action Editor“ ist nun der erste Keyframe zu sehen (siehe Abbildung 17).

Der zweite Keyframe soll den Schmetterling mit gesenkten Flügeln zeigen. Die Abwärtsbewegung der Flügel soll sich über zwanzig Frames erstecken. Daher wird der zweite Keyframe an der Stelle 21 eingefügt. Wieder wird zunächst der Frame gewechselt, um anschließend die Bones entsprechend zu rotieren und abschließend ein IPO Key einzufügen. Die Aufwärtsbewegung der Flügel soll sich ebenfalls über 20 Frames erstrecken. D. h. an der Stelle 41 sollen die Flügel des Schmetterlings wieder ihre ursprüngliche Ausgangslage erreicht haben.



Frame 1 Frame 21 Frame 41

Abbildung 18: Keyframes der "flying" Action.

Der Keyframe an der Stelle 41 soll identisch mit dem ersten Keyframe sein. Es bietet sich daher an, die Pose des ersten Keyframes zu kopieren und an der Stelle 41 wieder einzufügen. Um eine Kopie zu erstellen, muss zunächst in den Frame mit der Nummer eins gewechselt werden. Mit Hilfe von „Pose“ → „Copy Current Pose“ wird die Position der Knochen zwischengespeichert, und nachdem in den Frame mit der Nummer 41 gewechselt wurde, mit Hilfe von „Pose“ → „Paste Pose“ eingefügt.

Standardmäßig erwartet das XNA Framework, dass eine Animation 60 Frames umfasst. Da die vorliegende Animation nur 41 Frames beinhaltet, sind der Start und das Ende der Animation in der „Timeline“ festzulegen (siehe Abbildung 19). Mit Hilfe des „Play Timeline“ Buttons lässt sich die Animation nun abspielen.

Abbildung 19: Abspielen der Animation in der „Timeline“.

Die erstellte Aktion enthält Constraints, die in XNA nicht eingesetzt werden können. Mit Hilfe des „Bake“ Buttons im „Action Editor“ werden die Constraints in IPO-Kurven umgewandelt. Es entsteht eine Aktion, in der jeder Frame ein Keyframe ist. IPO-Keys und IPO-Kurven sind der Kern des Animationssystems in Blender. IPO-Kurven repräsentieren die Zwischenwerte für die Frames, in denen kein Key gespeichert ist. (12)

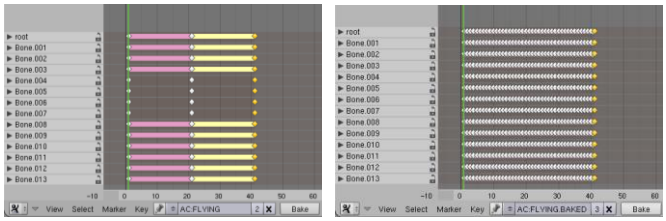


Abbildung 20: Action vor und nach dem "Bake".

Der Name der neuen Aktion besteht aus dem Namen der ursprünglichen Aktion („flying“) und dem Anhang „.BAEKED“, also im Beispiel „flying„.BAEKED“. Diese Aktion kann in XNA angesprochen werden. Wurde das „backen“ der Aktion nicht durchgeführt, gibt das XNA-Framework eine Fehlermeldung aus die besagt, dass das Input File keine Animationsdaten beinhaltet.

6 Export des Modells

Die neueste Version von Blender bietet zwar ein verbessertes Export-Skript für das FBX Format, jedoch genügt auch dieses Skript nicht den Anforderungen des XNA-Frameworks. Um die erstellten Animationen in XNA ansprechen und abspielen zu können, muss ein von „TripleB“ entwickeltes Skript verwendet werden. Das Skript kann unter „Triple B Games“ heruntergeladen werden (13).

Das Skript muss im Blender-Verzeichnis unter „Blender Foundation\Blender\.blender\scripts“ abgelegt

werden. Das Skript steht erst nach einem Neustart der Blender Anwendung zur Verfügung. Der Export des Modells erfolgt nun mit dem neuen Skript über „File“ → „Export“ → „Autodesk FBX (.fbx) Modified for XNA“.

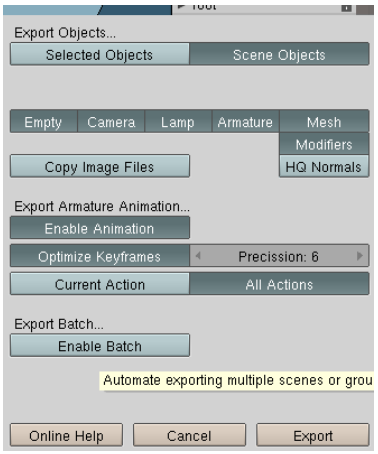


Abbildung 21: Export Einstellungen in Blender.

Für den Export des Modells sollten folgende Optionen, wie in Abbildung 21 dargestellt, ausgewählt werden: „Export Scene Objects“, „Optimize Keyframes“ und „All Actions“. Analog zum bisherigen Skript muss auch bei dieser Variante die exportierte FBX Datei manuell bearbeitet werden, - siehe hierfür (11).

7 Import des Modells

Wie eingangs erwähnt, bietet das XNA Framework standardmäßig nur eine eingeschränkte Unterstützung von Animationen. Das Objekt `Model` erlaubt es, Animationsdaten in der Content Pipeline zu speichern. Es beinhaltet aber keine Klassen, um diese Animationen zur Laufzeit abspielen zu können. Im Folgenden wird daher auf das SkinnedSample Projekt des XNA Crators Club zurückgegriffen. (4)

Die erstellte FBX Datei, wie auch die Textur des Modells werden zunächst dem Projekt „SkinningSample“ unter „Content“ hinzugefügt. Hierfür sollte innerhalb des XNA Game Studios, im Fenster „Projektmappen-Explorer“, das Kontextmenü des Zielverzeichnisses geöffnet werden. Mit Hilfe der Menüpunkte „Add“ → „Vorhandenes Element“ öffnet sich das Dialogfenster „Vorhandenes Element hinzufügen“. Evtl. ist es erforderlich, im Dialogfenster den Dateityp „Content Pipeline Files“ auszuwählen, damit die Modell- und Texturdateien in der Verzeichnisstruktur angezeigt werden.

An dieser Stelle unterscheidet sich der Import nun von der bisherigen Vorgehensweise. In den Eigenschaften des neuen Modells muss unter Content Processor nicht mehr der Eintrag „Model - XNA Framework“ sondern der Eintrag „SkinnedModelProcessor“ ausgewählt werden. Dieser Prozessor durchläuft das animierte Modell und erstellt Tags für die gefundenen Animationsdaten. (10)

Um die erstellte Animation abzuspielen wird das „dude“ Modell in der LoadContent Methode kurzerhand durch den neu hinzugefügten Schmetterling ersetzt:

```
// Load the model.
currentModel =
Content.Load<Model>("schmetterling");
```

Das Objekt `currentModel` ist noch wie vor vom Type Model, das XNA standardmäßig zur Verfügung stellt. Nachdem nun das Modell des Schmetterlings verwendet wird, muss auch die Methode für das Laden der Animation bzw. der Action aus Blender angepasst werden.

```
// Create an animation player, and start
decoding an animation clip.

animationPlayer = new
AnimationPlayer(skinningData);


AnimationClip clip =
skinningData.AnimationClips["flying.BAKE
D"];
```

Wird das Projekt nun erstellt und gestartet, erscheint ein flatternder Schmetterling auf dem Bildschirm.

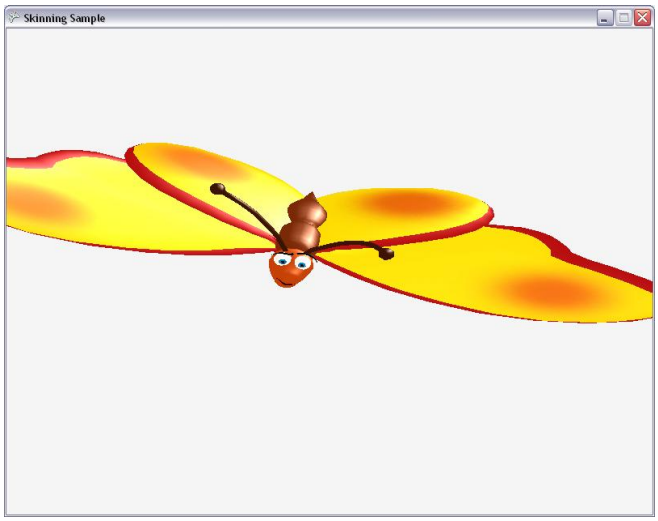


Abbildung 22: Animierter, gerenderter Schmetterling in XNA.

8 Zusammenfassung

Anhand eines durchgehenden Beispiels wurde gezeigt, wie sich im kostenfreien 3D-Modellierungstool Blender Animationen erstellen und in das kostenfreie XNA Game Studio einbinden lassen. Bei der Erstellung von Animationen für das XNA Framework ist darauf zu achten, dass eine zusammenhängende Armature erstellt wird. Ein weiterer wichtiger Punkt stellt das Weight Painting dar, bei dem absolut jedem Vertex

zumindest ein Bones zugewiesen werden muss. Die erstellten Actions müssen in Blender „gebacken“ werden, um von der XNA Game Engine gelesen werden zu können. Des Weiteren wurde ein Content Prozessor vorgestellt, der es ermöglicht, die erstellten Animationen in XNA abzuspielen.

Das gesamte Beispiel, einschließlich der Blenderdateien (.blend) vor und nach dem Erstellen der Animation, dem bearbeiteten FBX Format (.fbx), der Modelltextur (.png) sowie das XNA Game Studio Projekt, kann von der Webseite des Zentrums für Softwarekonzepte (ZfS) Karlsruhe heruntergeladen werden (7). Die Verwendung dieser Dateien ist gestattet, sofern folgender Urheberrechtshinweis hinzugefügt wird: „Copyright (c) 2008, Zentrum für Softwarekonzepte (ZfS) Karlsruhe“.

9 Literaturverzeichnis

1. **Langer, Stefan.** Animation - Proseminar: Computerspiele. *Otto-von-Guericke-Universität Magdeburg.* [Online] [Zitat vom: 20. 08 2008.] http://isgwww.cs.uni-magdeburg.de/~masuch/computerspiele/PS_Vortraege/Animation_in_Computerspielen.pdf.

2. **Autodesk, Corporation.** Autodesk FBX - Overview. *Autodesk GmbH.* [Online] 2008. [Zitat vom: 22. 08 2008.] http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=8224901.

3. **Microsoft, Corporation.** X File Format Reference. *MSDN - DirectX Developer Center.* [Online] 2008. [Zitat vom: 22. 08 2008.] http://msdn.microsoft.com/en-us/library/bb173014(VS.85).aspx.

4. —. Skinned Model. *XNA Creators Club Online.* [Online] 2008. [Zitat vom: 20. 08 2008.] http://creators.xna.com/en-us/sample/skinnedmodel.

5. **Blender, Foundation.** Get Blender. *Blender.* [Online] 2008. [Zitat vom: 22. 08 2008.] http://www.blender.org/download/get-blender/.

6. **Microsoft, Corporation.** Microsoft XNA Game Studio 2.0. *Microsoft Download Center.* [Online] 2008. [Zitat vom: 22. 08 2008.] http://www.microsoft.com/downloads/details.aspx?familyid=df80d533-ba87-40b4-abe2-1ef12ea506b7&displaylang=en.

7. **Fuchsloch, Andrea und Weidner, Katja.** Downloads ZfS Karlsruhe . *Zentrum für Softwarekonzepte (ZfS) Karlsruhe am Forschungszentrum Informatik.* [Online] 2008. [Zitat vom: 24. 08 2008.] http://zfs.fzi.de/zfs/Downloads/tabid/56/language/de-DE/Default.aspx.

8. **Blendercommuity, Deutschsprachige.** Das Pfefferkuchenmännchen. *Blender Dokumentation: Die erste Animation in 30 plus 30 Minuten.* [Online] 2008. [Zitat vom: 20. 08 2008.] http://de.wikibooks.org/wiki/Blender_Dokumentation:_Die_erste_Animation_in_30_plus_30_Minuten.

9. **Garstenauer, Martin.** Character-Animation in Echtzeit. *Johannes Kepler Universität Linz - Institute of Graphics and Parallel Processing.* [Online] 2008. [Zitat vom: 22. 08 2008.] http://www.gup.uni-linz.ac.at/~gk/Diplom/CAEDS1.pdf.

10. **Strom, Rick.** Modeling for XNA with Blender Part III. *Stromcode .* [Online] 13. 03 2008. [Zitat vom: 24. 08 2008.] http://www.stromcode.com/2008/03/13/modeling-for-xna-with-blender-iii/#more-92.

11. **Fuchsloch, Andrea und Weidner, Katja.** Blender to XNA Tutorial – Quick and Easy. *Zentrum für Softwarekonzepte (ZfS) Karlsruhe am Forschungszentrum Informatik.* [Online] 2008. [Zitat vom: 25. 08 2008.] http://zfs.fzi.de/LinkClick.aspx?fileticket=iIM6KGUU0lU%3d&tabid=131&mid=545&language=de-DE.

12. **Blendercommuity, Deutschsprachige.** Blender Dokumentation. *Das deutschsprachige Handbuch.* [Online] 2008. [Zitat vom: 04. 09 2008.] http://de.wikibooks.org/wiki/Blender_Dokumentation.

13. **Games, Triple B.** Script for exporting skinned anims to XNA from blender. *Triple B Games - Downloads.* [Online] 2008. [Zitat vom: 25. 08 2008.] http://www.triplebgames.com/downloads.html.