

# Blenderclonk Handbuch

Richard Gerum (Randrian)

20. Februar 2008

## Inhaltsverzeichnis

<b>1</b>	<b>Wichtige Dateien</b>	<b>2</b>
<b>2</b>	<b>Clonk laden</b>	<b>3</b>
<b>3</b>	<b>Hauptmenü</b>	<b>3</b>
<b>4</b>	<b>Action</b>	<b>4</b>
<b>5</b>	<b>Clonk</b>	<b>4</b>
<b>6</b>	<b>Save/Load</b>	<b>5</b>
<b>7</b>	<b>Render</b>	<b>5</b>
<b>8</b>	<b>Tools</b>	<b>6</b>
<b>9</b>	<b>Importieren</b>	<b>7</b>
<b>10</b>	<b>Beispiel</b>	<b>8</b>
10.1	Der Bauer Humpe will Clonk werden . . . . .	8
10.2	Bauer sucht Hut . . . . .	8
10.3	Humpe will eine Hose haben . . . . .	9
10.4	Eine Mistgabel für Mr. Humpe . . . . .	10
10.5	Bauernaufstand! Humpe kämpft . . . . .	11



# 1 Wichtige Dateien



Natürlich wird Blender benötigt!

Ohne Blender geht hier nichts. Wer also noch kein Blender hat sollte es sich deswegen erst von <http://www.blender.org/> runterladen und installieren.

Blender installiert? Dann kann es ja weitergehen...

Die wichtige Hauptdatei, ist **Clonk.blend**. In ihr sind das Armatureobjekt (Skelett des Clonkes) und die Scripte gespeichert.

Zum Erstellen der Grafiken wird **Anigrab.exe** benötigt, das im gleichen Verzeichnis liegen sollte wie die .blend Datei.

Um die Scripte richtig aus zu führen wird eine Pythoninstallation benötigt. Diese kann von <http://www.python.org/> geladen werden.

Es gibt dann noch vier wichtige **Unterordner**:

Im Ordner **Actions** sind alle Animationen, die für die einzelnen Clonks gebraucht werden als .anim Datei abgelegt.

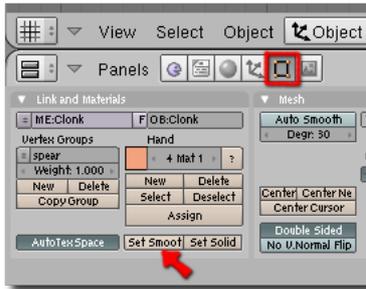
Der Ordner **Clonks** enthält .mesh Dateien, in denen die Meshes der einzelnen Clonks gespeichert sind. In den jeweiligen .act Dateien ist dann zu dem Clonk die Abfolge der Actions gespeichert, in der Reihenfolge wie sie dann später in der Graphics.png erscheinen soll.

Im Ordner **Tools** sind die Tools für die Animationen gespeichert. Das sind die Werkzeuge, die der Clonk bei verschiedenen Animationen verwenden soll, wie z.B. die Schaufel oder den Hammer.

Im Ordner **Graphics** werden dann nach dem Rendern die Graphics.png Dateien für die Clonks abgelegt.

## 2 Clonk laden

Um den Renderclonk zu starten muss die Datei Clonk.blend geöffnet werden. Jetzt sieht man unten rechts in der Ecke die Scriptgui (Benutzeroberfläche des Scriptes). Dort kann man nun mit „**Load Clonk**“ einen Clonk aus dem Unterordner Clonks auswählen. Mit dem Knopf daneben  kann man auch schnell einen Clonk mit Hilfe eines Menüs auswählen. „**Import Mesh**“ hilft beim Importieren eines Clonkes (siehe Kapitel Importieren). „**Quit**“ beendet Blender.



Wurde nun ein Clonk geladen, empfiehlt es sich zunächst **SetSmooth** zu machen. Dazu den Clonk anwählen (rechtsclick aufs Mesh). Dann in dann auf  gehen und dann auf auf SetSmooth klicken.

Da diese Einstellung leider per Script nicht setzbar ist muss dies bei jedem Laden erneut geschehen. Doch besonders beim Picturebild kann man deutliche Unterschiede zwischen SetSmooth und SetSolid (dem Standard) erkennen.

## 3 Hauptmenü

Mit Hauptmenü gibt es jetzt folgende Schaltflächen:



**Action:** Hier können verschiedene Einstellungen zur Action gemacht werden.

**Clonk:** Hier wird die Actionliste für den Clonk bearbeitet.

**Save/Load:** Hier kann man Actions und Tools speichern und laden.

**Render:** Dient zum Rendern des Clonkes.

**Exit:** Zurück zum vorherigen Menü zum Laden eines anderen Clonkes.

**Achtung:** der Clonk wird dabei nicht gespeichert, eventuelle Änderungen gehen verloren!

Dazu gibt es noch eine Eingabezeile in der der Pfad für den temporären Ordner angegeben ist und dort geändert werden kann.

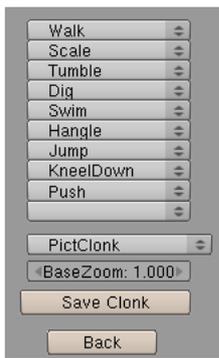
## 4 Action



Mit dem Dropdownmenü ganz oben können verschiedene **Actions** ausgewählt werden. Hat man das Armatureobjekt markiert erschienen dazu im Actionfenster dann die Keyframes der einzelnen Bones. In den unteren Feldern kann man die **Länge (Length)** der Action einstellen, also wie viele Frames die Action lang ist, und die **Breite (X)** und **Höhe (Y)** der einzelnen Framegrafiken. Zudem kann man noch zwei **Werkzeuge (Tools)** festlegen, die der Clonk bei dieser Action verwenden soll. Diese werden dann jeweils auf den Bone 'Tool1' bzw. 'Tool2' gelegt.

Mit **Rename** kann die Action umbenannt werden. Mit **New** wird eine neue Action hinzugefügt, wahlweise mit alten Keyframes oder ohne. Der untere Knopf ist dann zum **Speichern** der Action. Dies sollte man immer tun, wenn man Keyframes oder Einstellungen zur Action verändert hat. Die Action wird dann im Unterordner Actions gespeichert.

## 5 Clonk



Hier sieht man die Liste der Actions des Clonkes. Wenn mehrere Spalten zu sehen sind, enthalten die rechten Spalten immer die Actions, die vor denen kommen, die in der Spalte links davon zu sehen sind.

Da oft der Platz nicht ausreicht um alle Actions zu sehen kann man das Fenster mit einem Klick auf  auf den ganzen Bildschirm vergrößern. Mit einem erneuten Klick auf diese Schaltfläche kann man das Fenster wieder verkleinern.

Zum nun die Liste zu verändern muss kann man auf eine Action klicken und dann statt dessen eine andere auswählen. Mit **REMOVE** kann man die Action aus der Liste löschen. Mit **INSERT** wird über der Action ein Platz eingefügt. Die Actions mit einem

U davor sind Actions, die in der Liste noch nicht vorkommen.

Die abgesetzte Box am unteren Ende der Liste enthält die Action für das **Picturebild** des Clonks.

Darunter kann man den **BaseZoom** einstellen. Das ist die Zoomeinstellung der Kamera, und wird benötigt falls man ein Mesh hat das zu groß ist und kleiner dargestellt werden soll, bzw andersherum. Diese Einstellung wird aber normalerweise bei den Clonks nicht benötigt.

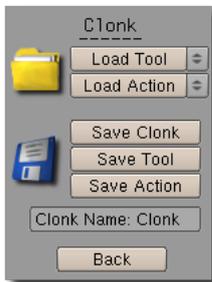
Der unterste Schalter **speichert** die Actionliste und das Mesh des Clonkes.

**Achtung:** Die Clonk dürfen immer nur aus **einem** Mesh bestehen! Hat man zwei Meshes können diese mit Strg+J verbunden werden.

Auch muss immer über das Scriptinterface gespeichert werden und nicht mit der normalen Blender-Speicherfunktion!

Dabei werden jedoch exotische Einstellungen nicht exportiert! Also wenn nach dem Laden z.B. ein bestimmter Modifier oder eine Textureinstellung nicht aktiviert ist, liegt es daran, das diese nicht unterstützt wird.

## 6 Save/Load



Hier können Actions und Tools geladen werden. Am Anfang beim Laden des Clonks werden nur die Actions und Tools geladen, die der Clonk benötigt. Falls man weitere Actions von anderen Clonks braucht kann man diese hiermit laden. Dabei kann man mit einem Klick auf den Schalter sich die **Action/Tool** mit dem Menüsystem aussuchen oder mit einem Klick auf  direkt aus einem Dropdownmenü aussuchen.

Dieses Menü dient auch zum Speichern. Man kann hier eine **Action**, ein **Tool** und den **Clonk** speichern. Action und Clonk können auch in den anderen Untermenüs (Action bzw. Clonk) gespeichert

werden. Tools können nur hier gespeichert werden.

Das Eingabefeld ganz unten dient zum ändern des **Clonknamens**. Hat man z.B. den normalen Clonk geladen und bearbeitet kann man ihn hier vor dem Speichern umbenennen, damit der normale Clonk nicht überschrieben wird.

**Achtung:** die normale Speicherfunktion von Blender kann hier NICHT zum speichern verwendet werden. Zum Speichern bitte immer die Scriptgui (das Fenster unten rechts) verwenden.

## 7 Render



Mit einem Klick auf die Renderschaltfläche öffnet sich ein Menü. Hier können verschiedene **Actions** ausgewählt werden, die separat gerendert werden sollen. Mit **-All-** kann der gesamte Clonk durchgerendert werden.

Mit weiteren Untermenü kann man dann auswählen ob man mit oder ohne

Overlay gerendert haben möchte. **Both** rendert dabei beides, Graphics.png und Overlay.png. **Graphics/Overlay** rendern jeweils nur eines von beiden. Mit **Full** kann man dann beides in eine Grafik rendern. Dies dient dazu dass man sich den Clonk nach dem Rendern besser anschauen kann.

Beim Rendern wird jeder Einzelframe im temporären Verzeichnis gespeichert. Diese werden dann nach dem Rendern mit **Anigrab** zur Grafik zusammengebaut. Diese findet sich dann im Ausgabeordner /Graphics/Clonkname/Graphics.png. Dabei werden die Actions in der **Reihenfolge**, wie sie in der Liste vorkommen in die Grafik gestellt. Die Breite der Grafik richtet sich dabei nach der breitesten Action. Passen mehrere Actions nebeneinander werden sie auch so in die Grafik gestellt. Zum Schluss wird das **Picture** an einen freien Platz oder das Ende der Grafik gestellt. Damit man die ActMap zu dem Clonk besser erstellen kann werden die Positionen der Actions in einer Datei im Ausgabeordner gespeichert.

Wird nur eine **einzelne Action** gerendert, dann wird trotzdem versucht die gesamte Grafik zu bauen. Es sollten also von den anderen Actions schon gerenderte Frames in dem temporären Ordner liegen. Diese Funktion dient dazu, dass man nicht immer alle Actions neu rendern muss, wenn man nur an einer Action etwas verändert hat. Zu Anfang muss jedoch einmal komplett durchgerendert werden, damit die Grafiken der anderen Actions schon im temporären Ordner liegen.

## Anmerkung

Generell werden alle Materialien, die mit **Overlay\*** anfangen, bei **Graphics** nicht gerendert. Bei **Overlay** werden nur die Materialien mit **Overlay\*** gerendert. Bei **Full** werden beide gerendert. Zusätzlich wird bei Full noch das Material **OverlayColor** blau statt weiß gefärbt.

Das Material **Face** wird in den normalen Animationen mit der Textur **'Face'** belegt und beim Picture mit **'Face2'** geskinnt. Dies ist nötig, da das hochauflösende Gesicht in der kleinen Animation untergehen würde und deswegen eine gröbere Textur verwendet werden muss.

## 8 Tools

Tools sind die **Werkzeuge** des Clonks, die er bei verschiedenen Animationen verwendet. Die Tools des Standartclonks sind Schaufel, Axt, Hammer und Flasche. Beim Ritter kommen dann noch diverse Waffen dazu.

Um ein neues Tool zu erstellen muss zunächst ein **passendes Mesh** gebaut werden. Wichtig hierbei ist auch, dass das Tool nur aus **einem** Mesh besteht. Hat man ein Mesh modelliert muss es noch passend platziert werden, damit es später bei der Verwendung auch richtig am Bone sitzt. Dazu setzt man am besten die Action 'Empty' für das Armatureobject (oder eine andere Animation, die keine Tools verwendet). Jetzt sieht man über dem Kopf des Clonkes die beiden Toolbones auf einer Stelle. Das Mesh muss nun so platziert werden, dass die Stelle, an der der Clonk das Tool anfassen soll direkt auf dem Bone zu liegen kommt. Das ganze am besten noch von verschiedenen Ansichten prüfen.

Nun sollte das neue Tool noch passende benannt werden. Dabei sollte **Objectname** und **Meshname** identisch sein.



Jetzt kann das Tool über **Save/Load** und **'Save Tool'** gespeichert werden. Dazu dann das Tool aus der aufklappenden Meshliste auswählen. Das Tool wird nun unter `/Tools/toolname.mesh` gespeichert.

## 9 Importieren

Um einen Clonk zu importieren muss auch erst Clonk.blend geladen werden. Jetzt brauchen wir das zu importierende Clonkmesh. Dieses kann über **File->Append or Link** aus der alten Datei geladen werden. Dazu die Datei in dem geöffneten Menü auswählen, dann auf Objects und dann das Objekt wählen, das den Clonk enthält. Wichtig ist, dass der geladene Clonk nur aus einem Mesh besteht! Besteht er aus mehreren müssen diese zunächst mit beide selektiert (Shift+Rechtsklick) und mit Strg+J verbunden werden. Das neue Mesh muss jetzt noch auf den Namen des Clonkes umbenannt werden.



Dabei sollte Objectname und Meshname identisch sein. Danach kann das selektierte Mesh mit einem Klick auf **'Import Mesh'** importiert werden. Im Untermenü Clonk können dann die Actions für den Clonk angepasst werden und der Clonk gespeichert werden. Jetzt kann man ihn genauso wie die anderen Clonks

verwenden.

## 10 Beispiel

Wer sich noch nicht mit dem Modellieren in Blender auskennt, sollte zuerst ein allgemeines Blendertutorial machen!

Dazu am besten unter <http://www.blender.org/education-help/tutorials/> eines der Anfangstutorials durchgehen. Die Blender.org Tutorials sind allerdings in englisch.

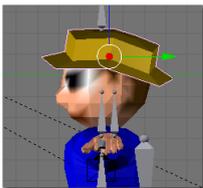
Oder Blendertutorial vom Clonk-Center: z.B. [ein Flint](#), [ein Bausatz](#) oder [ein Schwert](#)

### 10.1 Der Bauer Humpe will Clonk werden

Wir wollen nun einen Bauern Clonk basteln. Dazu muss erstmal Blender gestartet und das Script geöffnet werden. Da der Bauer auf dem normalen Clonk bassieren soll muss erstmal dieser geladen werden. Also im Scriptfenster auf **LoadClonk** oder  klicken und Clonk.mesh (bzw. Clonk) auswählen. Jetzt befinden wir uns im Hauptmenü und haben den Clonk geladen. Dieser soll jetzt aber umbenannt werden zum Bauer, damit wir nicht den normalen Clonk überschreiben. Dazu gehen wir auf **Save/Load** und geben unter **Clonk Name** statt Clonk nun Farmer ein (es ist allgemein üblich die Clonks in englisch zu bezeichnen, natürlich würde Bauer auch gehen). Nun kann der Bauer mit einem Klick auf **Save Clonk** gespeichert werden. Gut, wir haben jetzt schon einen neuen Clonk geschaffen: den Bauer!

### 10.2 Bauer sucht Hut

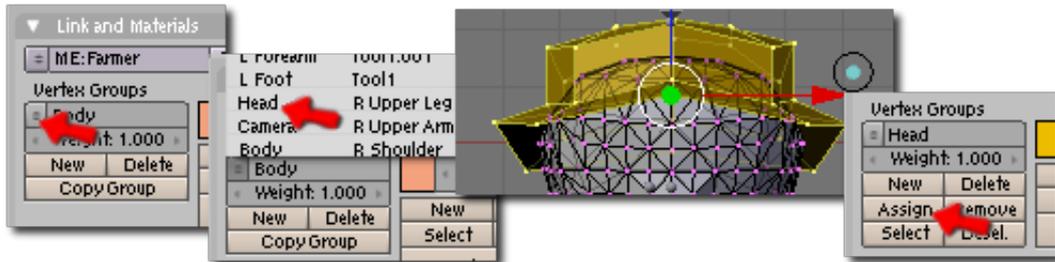
Damit man den Unterschied zwischen Bauer und Clonk erkennt sollte der Bauer natürlich anders aussehen. Bauer Hump möchte gerne einen Hut auf haben. Also geben wir ihm einen.



Zuerst muss dazu ein passen der Hut modelliert werden. Das sollte mit einem grundlegenden Blendertutorial wie den oben genannten kein großes Problem darstellen.

Der Hut ist zunächst ein einzelnes Mesh. Aber damit er zusammen mit dem Clonk gespeichert werden kann muss er mit dem Clonkmesh verbunden werden. Dazu muss im ObjektMode (gegebenenfalls aus dem EditMode mit Tab gewechselt werden) zuerst das Hutmesh mit rechts angeklickt werden und dann während man Shift gedrückt hält noch das Clonkmesh mit rechts angeklickt werden. Jetzt hat der Hut einen dunklen Rand und der Clonk einen hellen. Mit Strg+J können die beiden Meshes nun verbunden werden.

Damit der Hut immer passend mit dem Kopf bewegt wird muss er jetzt eine passende VertexGroup bekommen. Dazu auf  gehen, dort links unter „Vertex Group“ auf das Symbol neben dem Textfeld gehen. Aus dem Dropdownmenü dann 'Head' (für das Head Bone, also den Knochen der den Kopf steuert) auswählen. Danach mit Tab in den Edit Mode wechseln, alle Vertices des Hutes (es können auch ein paar vom Vertices vom Kopf mit drin sein, das macht nichts) markieren und mit einem Klick auf Assign der Vertexgroup zuweisen.

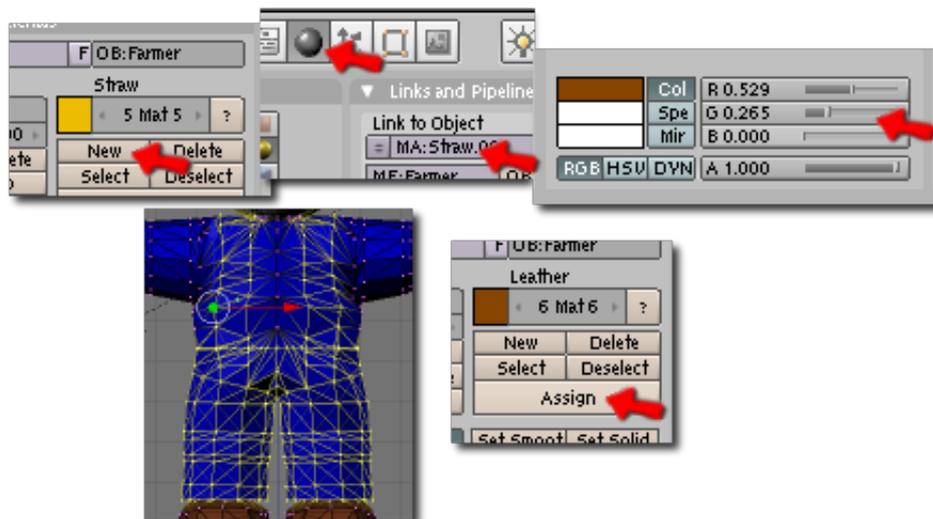


Im Scriptfenster kann nun unter **Clonk** dann mit **Save Farmer** der Bauer mit seinem Hut gespeichert werden.

### 10.3 Humpe will eine Hose haben

Als nächstes will Herr Humpe eine tolle Hose haben, denn der Einheitsoverall der Clonks ist für agrarwirtschaftliche Zwecke nicht adequat. Also muss eine braune Lederhose her.

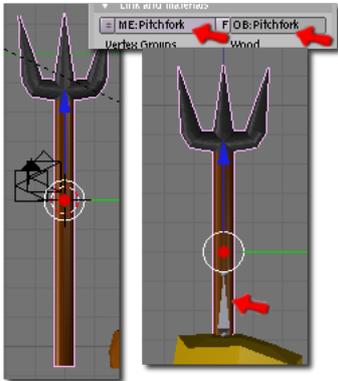
Dazu muss zuerst eine neue Materialgroup her. Dazu unter  im Bereich rechts neben den Vertexgroups auf 'New' klicken. Dann auf  und dem neu erzeugten Material einen Namen geben, hier bietet sich 'Leather' für ein Ledermaterial an. Mit den Farbwähler dann einen passenden Brauntönen einstellen. Nun muss das Material noch den passenden Stellen am Clonk zugewiesen werden. Dazu im EditMode die Vertices markieren, die die Lederhose umspannen sollen. Dann unter  auf Assign gehen, damit das Material den selektierten Faces zugewiesen wird.



So Humpe hat nun seine Hose an. Jetzt wird es mal wieder Zeit zu speichern.

## 10.4 Eine Mistgabel für Mr. Humpe

Als Bauer braucht man natürlich eine Mistgabel, sonst ist man ja kein richtiger Bauer!



Dazu muss natürlich zuerst einmal die landwirtschaftliche Gabel modelliert werden. Das Mistgabel-Mesh sowie das Object sollte dann den Namen 'Pitchfork' (eng. für Mistgabel) gegeben werden. Dann muss das Mesh so verschoben werden, dass die Position, an der der Clonk später die Gabel anfassen soll genau auf dem Tool1 Bone es Armatureobjectes ist (siehe Bild). Nun kann die Gabel über 'Save/Load' mit dem Button 'Save Tool' gespeichert werden, dazu im aufklappenden Menü den Eintrag 'Pitchfork' auswählen.

Jetzt soll Humpe die Gabel natürlich beim Laufen immer tragen. Dazu werden wir als Grundlage die Action 'WalkStaff' verwenden. Diese muss zunächst geladen werden. Dazu gehen wir auf 'Save/Load' und wählen den Eintrag 'Load Action' aus und laden die Action 'WalkStaff'. Jetzt geht's ins Untermenü 'Actions' da wählen wir die Action 'WalkStaff' aus. Mit einem Klick auf 'New' wird eine neue Action erstellt. Als neuen Namen geben wir 'WalkPitchfork' ein. Die Frage nach dem Kopieren der Keyframes beantworten wir mit ja. Jetzt haben wir also eine neue Action mit dem Namen Pitchfork und den Keyframes von WalkStaff. Jetzt muss noch als Tool1 Pitchfork eingestellt werden, damit der Clonk statt dem Stab die Mistgabel verwendet. Mit einem Klick auf 'Save WalkPitchfork' kann die neue Action nun gespeichert werden.

Damit mit der neuen Action statt der alten Walk Animation gerendert wird müssen wir die noch in der Actionliste des Clonks umstellen. Dazu gehen wir auf 'Clonk' und machen das Fenster mit  einem Klick auf groß. Jetzt können wir 'Walk' anklicken und statt dessen 'WalkPitchfork' auswählen.

Was gleiche können wir jetzt für die Jump Action machen. Dazu laden wir 'JumpStaff'. Mit JumpStaff wird dann die neue Action JumpPitchfork erzeugt und Pitchfork als Tool1 zugewiesen, dabei nicht vergessen die Länge der Jumpanimation auf 8 Frames statt 16 zu stellen. Jetzt wird Jump in der Actionliste durch JumpPitchfork ersetzt.

Jetzt läuft und springt Humpe glücklich mit seiner Mistgabel! Zeit zu speichern.

Nun kann Bauer Humpe endlich gerendert werden und sich seiner Clonkheit freuen:



## 10.5 Bauernaufstand! Humpe kämpft

Doch das alles ist noch nicht genug! Humpe will seine Mistgabel auch zum Kämpfen verwenden.

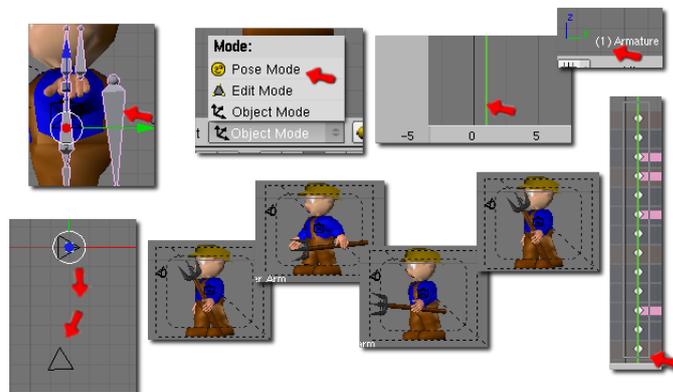
Also müssen Kampfanimationen her. Da man aber mit einer Mistgabel nicht einfach zu zuschlägt wie mit einem Schwert können wir nicht die Schwertanimation nehmen und die Mistgabel mit dem Schwert austauschen. Folglich müssen wir eine ganz neue Animation machen.

Dazu gehen wir zunächst ins Untermenü Action und klicken auf 'New' und geben als neuen Namen den Namen PitchforkStrike ein. Die Frage nach dem Kopieren von Keyframes verneinen wir. Die Animationslänge stellen wir auf 8 und die Breite auf 24. Die Höhe von 20 belassen wir so. Jetzt muss das Armatureobjekt angewählt werden und in den 'Pose Mode' gewechselt werden.

Zunächst muss in den Frame 1 gewechselt werden. Die Framenummer wird immer links unten im 3D Fenster angezeigt, bzw. als grüner Stich im Animationsfenster (siehe Bild). Dazu entweder im Animationsfenster auf die Position vom Frame 1 klicken oder mit den Pfeiltasten die Frames durchschalten. Sind wir nun im Frame 1 angekommen können wir die Bones des Clonkes so verbiegen, wie er am Anfang der Angriffsanimation aussehen soll. Dabei sollte die Kamera von schräg vorne auf den Clonk blicken, weil das bei den meisten Animationen so ist.

Fertig positionierte Bones müssen dann noch mit einem Klick auf die Taste 'i' und der Auswahl von 'LocRot' in die Keyframeliste übernommen werden. Ist die Pose fertig und alle Bonepositionen als Keyframes gespeichert kann es weitergehen. Dann geht es weiter mit Frame 5, hier soll der Clonk gerade beim Ausholen ganz hinten sein (die zwischen Frames werden von Blender dann automatisch berechnet). Im Anschluss geht es mit Frame 7 weiter, hier soll der Clonk gerade ganz nach vorne stechen. Als 9. Frame wollen wir wieder den ersten haben, damit der Clonk wieder in die Ausgangsstellung zurückkehrt. Also drücken wir 'b' um im Actionfenster alle Keyframes des ersten Frames nun mit einem gezogenen Rahmen zu markieren. Mit Shift+D werden diese nur dupliziert und an die Stelle von Frame 9 verschoben. Warum Frame 9, den sieht man doch gar nicht, wir haben ja nur 8 Frames! Wir brauchen den 1. Frame ja nicht doppelt in der Animation, diese Kopie dient nur dazu das der Frame 8 als Übergang von 7 zu dem 1 von Blender berechnet wird.

Ein Klick auf 'Save StrikePitchfork' speichert nun die neue Action.



Jetzt brauchen wir natürlich auch noch eine Animation in der Humpe mit der Mistgabel in Kampfstellung ist (dann wenn sich zwei Clonk im Kampf gegenüberstehen). Hier reicht eine Kopie von SwordFight. Also laden wir SwordFight über 'Save/Load' und 'Load Action'. Im 'Action' Untermenü kann jetzt mit 'New' eine Kopie der Action erstellt werden. Da muss jetzt die Länge auf 4 und die Breite auf 24 gesetzt werden. Also Tool1 stellen wir wieder unsere Pitchfork ein. Jetzt kann man jedoch erkennen, dass der Clonk die Mistgabel zu hoch hält. Also im 'Pose Mode' das Tool1 Bone etwas tiefer setzen und einen Keyframe mit 'i' und 'LocRot' einfügen. Das ganze bei Frame 1, 3 und 5 jeweils so machen. Mit 'Save PitchforkFight' wird die Action dann gespeichert.

Jetzt müssen die beiden Actions noch in die Actionliste des Clonks aufgenommen werden. Dazu auf das leere Feld unter Drink klicken und 'PitchforkFigth' auswählen. Dannach in das nächste leere Feld das dann unter 'PitchforkFight' entsteht klicken und 'PitchforkStrike' auswählen.

Ein Klick auf 'Save Farmer' speichert und unsre Zusammenstellung.

Jetzt ist unser kleiner Agrarwirt endlich fertig. Mit einen Klick auf 'Render', 'All' und dann 'Both' wird nun Graphics.png und Overlay.png im Unterordner Graphics/Farmer erzeugt. Von dort können sie jetzt ins Spiel eingebaut werden.

