

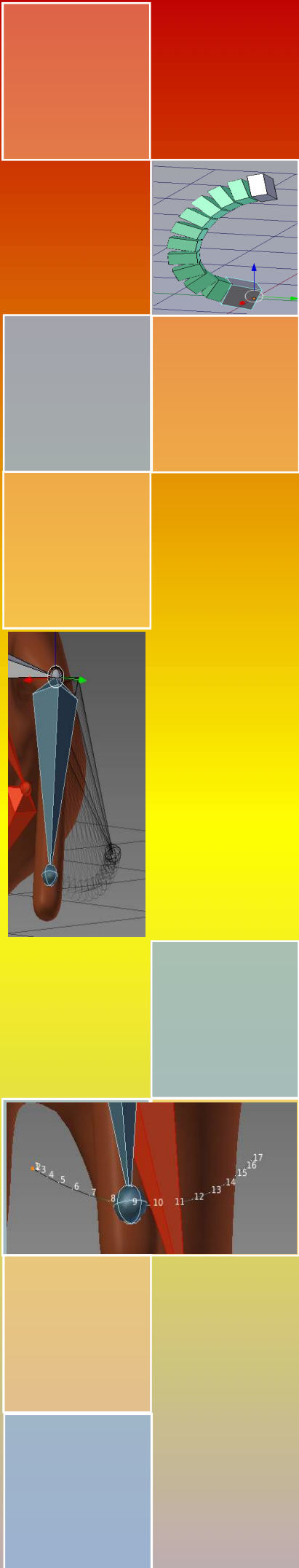
# Blender – Das Handbuch

## Kapitel 7: Rigging und Animation mit Armature

---

Henricus

Version 05. August 2017



# Inhalt

|        |  |    |
|--------|--|----|
| 7      | Rigging und Animation mit Armature                 | 3  |
| 7.1    | Basics   | 3  |
| 7.1.1. | Begriffe   | 3  |
| 7.1.2. | Arbeitsschritte                                    | 3  |
| 7.1.3. | Darstellungsformen von Bones                       | 4  |
| 7.1.4. | Knochen und Knochenketten                          | 5  |
|        | Object Mode  | 5  |
|        | Edit Mode  | 5  |
| 7.1.5. | Pose Mode  | 6  |
|        | Beispiel 1: Verbindung der Armature mit einem Mesh | 7  |
| 7.1.6. | Bendy Bones  | 8  |
|        | Beispiel 2   | 8  |
| 7.2    | Einstellungen der Armatures                        | 11 |
|        | Skeleton   | 11 |
|        | Display  | 11 |
|        | Bone Groups  | 12 |
|        | Pose Library                                       | 12 |
|        | Ghost  | 13 |
|        | Inverse Kinematics                                 | 13 |
|        | Motion Paths                                       | 13 |
| 7.3    | Einstellungen der Bones                            | 14 |
|        | Transform  | 14 |
|        | Transform Locks                                    | 14 |
|        | Relations  | 14 |
|        | Display  | 15 |
|        | Inverse Kinematics (IK)                            | 15 |
|        | Deform   | 15 |
| 7.4    | Bone Constraints                                   | 16 |
| 7.4.1  | Transform Bone Constraints                         | 16 |
|        | Copy Location                                      | 16 |
|        | Copy Rotation und Copy Scale                       | 17 |
|        | Copy Transforms                                    | 18 |
|        | Limit Distance                                     | 18 |
|        | Limit Location / Rotation / Scale                  | 18 |
|        | Maintain Volume                                    | 18 |
|        | Transformation                                     | 19 |
| 7.4.2  | Tracking Bone Constraints                          | 21 |
|        | Clamp To   | 21 |

|   |    |
|---|----|
| Damped Track.....                         | 21 |
| Inverse Kinematics .....                  | 21 |
| Locked Track.....                         | 21 |
| Spline IK.....                            | 22 |
| Stretch To.....                           | 23 |
| Track To.....                             | 23 |
| 7.4.3 Relationship Bone Constraints ..... | 24 |
| Action.....                               | 24 |
| Shrinkwrap .....                          | 26 |
| 7.4.4 Inverse Kinematics .....            | 26 |
| Der Grundgedanke .....                    | 26 |
| Automatic IK .....                        | 27 |
| Inverse Kinematic (IK).....               | 27 |
| Bewegungsmöglichkeiten einschränken ..... | 28 |

## 7 Rigging und Animation mit Armature

### 7.1 Basics

#### 7.1.1. Begriffe

*Armature* meint im Englischen (neben verschiedenen anderen Bedeutungen) das Gerüst einer Skulptur, an dem deren Außenhülle befestigt ist. Bei Blender geht es, wenn von Armatur gesprochen wird, um die Ausstattung von Meshes mit einem Knochengerüst, dessen einzelne Knochen (Bones) in bestimmte Richtungen bewegt werden können und die dabei Teile des Modells deformieren, indem sie diese quasi „mitnehmen“.

*Rigging* kann ebenfalls verschiedene Bedeutungen haben, wobei *manipulieren* in unserem Fall wohl die treffendste ist. Im Puppentheater meint *Rigg* nämlich das Gestell, an dem die Puppe mit Schnüren aufgehängt ist, um sie damit zu bewegen. Bei Blender lassen sich also mit Hilfe der Armatur Modelle von Menschen, Tieren oder auch Maschinen manipulieren.

Es sei aber angemerkt, dass *Rigging* bei Blender in manchen Zusammenhängen auch im Sinne von „*ein Mesh mit einem Knochengerüst versehen*“ verwendet wird.

Einige Aspekte des Riggings wurden bereits in Kap. 3.4.2 bei der Besprechung von **Armature**, einem **Deform Modifier**, erläutert. Jetzt werden Prinzipien noch einmal ausführlicher dargestellt.

#### 7.1.2. Arbeitsschritte

Rigging läuft (normalerweise) in dieser Reihenfolge ab:

1. **Modell erschaffen:** Rigging setzt voraus, dass ein Mesh existiert, das mit Knochen (Bones) versehen wird, die dann dieses Mesh in irgendeiner Weise bewegen. Dieses Modell kann Mensch, Tier oder Alien sein, aber auch eine Maschine, Roboter oder Automobil. Auch Pflanzen können durch Bones bewegt werden.
2. **Bones bzw. Armature einfügen:** In das Modell werden nun an geeigneten Stellen Knochen eingefügt. Wie du das machst, hängt einerseits von der „Natur“ des Modells ab; z.B. wird man an einem Arm die Bones so einfügen, dass man diesen in seinen Gelenken bewegen kann. Andererseits hängt das von der Genauigkeit ab, mit der dein Mesh modelliert ist. Wenn dein Alien nur drei Finger hat, brauchst du dafür vielleicht auch nur drei Bones. – Mit den AddOns **Human (Meta-Rig)** und **Pitchjoy Human (Meta-Rig)** können ganze (menschliche) Skelette eingefügt werden, mit denen allerdings nicht ganz einfach umzugehen ist.
3. **Bones im Detail Ausrichten:** Vor allem bei organischen Modellen ist diese Nacharbeit erforderlich (z.B. bei den einzelnen Fingergliedern).
4. **Skinning (Verbindung des Modells mit der Armature):** Dabei handelt es sich im Prinzip um eine Form des *Parenting*, wobei das Mesh das Child und die Armature der Parent ist. Das Besondere dabei ist, dass hier einzelne Knochen nur auf die sie umgebenden Teile des Mesh Einfluss haben, nicht auf das gesamte Mesh. Skinning kann auf verschiedene Weise erfolgen. Am einfachsten ist i.d.R. die Option **With Automatic Weights**. Skinning bedeutet immer auch die Festlegung des Einflussbereichs der Bones im Mesh.
5. **Constraints insbes. Inverse Kinematics einfügen:** Die Bones einer Armature sind immer in irgendeiner Weise miteinander verbunden und haben daher Einfluss aufeinander. Wie sich dieser Einfluss im Detail auswirkt, wird durch sog. Constraints festgelegt.
6. **Bewegungsmöglichkeiten einschränken:** Viele Gelenke von Menschen, Tieren und Maschinen lassen sich nicht in alle Richtungen bewegen. Darum ist es sinnvoll, die Bewegungsmöglichkeiten von vornherein so einzuschränken, dass keine unzulässigen Posen zustande kommen *können*.
7. **Handles (Anfasser) einfügen:** Besonders wenn ein Rig aus sehr vielen Bones besteht,

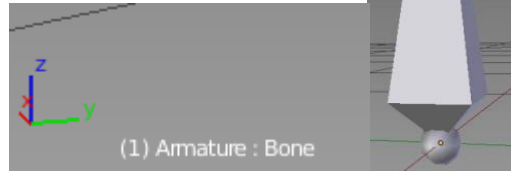
ist es sinnvoll, die Armature mit Anfassern zu versehen. Die einzelnen Bones müssen dann überhaupt nicht mehr bewegt werden.

### 7.1.3. Darstellungsformen von Bones

4

- Einen einzelnen Bone fügst du so ein: [shift] + [A] → **Armature** → **Single Bone**. Du erhältst dann dieses Gebilde:

- An dem breiten Ende befindet sich der *Kopf (Head)*, am schmalen Ende der Schwanz (*Tail*), am; diese beiden Teile können im Edit Mode separat mit RMT angewählt werden. Klickt man auf den Mittelteil, ist der ganze Bone ausgewählt.
- Im Edit Mode ist links unten im 3D View angezeigt **(1) Armature : Bone**. Das bedeutet: Das ist ein Teil (nämlich der Bone) von dem Objekt Armatur (das aus mehreren Bones bestehen kann).



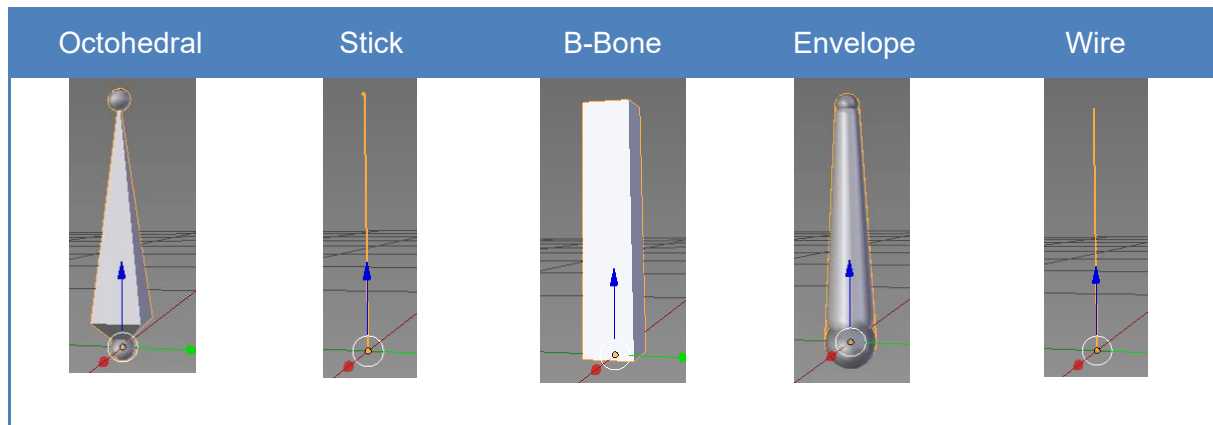
- Gleichzeitig mit dem Aufruf eines Bones erscheinen im Properties Editor drei neue Icons, die mit der Armatur zu tun haben:

- Das Männchen bezieht sich auf die **Object Data** der **Armature**.
- Der Knochen auf einzelne **Bones**.
- Knochen und Kettenglied beziehen sich die **Bone Constraints**.



Die Details hierzu werden noch erläutert.

- Wenn du die **Object Data** aufrufst (LMT auf das Männchen), findest du im Untermenü Display fünf Möglichkeiten für Formen, die ein Knochen haben kann:



Wenn der jeweilige Knochen selektiert ist, wird er durch Klick mit LMT auf eine der Schaltflächen in die jeweilige Form umgewandelt.

- Im Menü **Bone** (Knochen-Icon) findest du das Untermenü **Display**. Dort gibt es weitere Möglichkeiten, das Erscheinungsbild von Knochen zu verändern:
  - **Hide**: Setzen eines Häkchens bewirkt, dass der Knochen unsichtbar wird.
  - **Custom Shape**: Nach Klick auf das darunter stehende Feld kannst du aus den in der Szene vorhandenen Meshes eines auswählen, dessen Form der Bone annehmen soll. Auf diese Weise können Bones jede beliebige Form erhalten.
  - **Wireframe**: Diese Funktion ist nur verfügbar, wenn zuvor Custom Shape verwendet wurde. Die neue Form ist dann als Drahtgitter sichtbar.

- Der *B-Bone* (s.o., mittleres Bild) ist nicht nur eine andere Darstellungsform, sondern diese Einstellung ist auch Voraussetzung dafür, dass du mit Bendy Bones (s. nächstes Unterkapitel, 3. Abschnitt) arbeiten kannst.

## 7.1.4. Knochen und Knochenketten

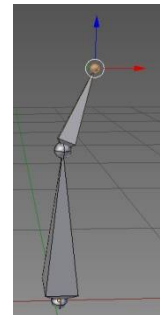
### Object Mode

- Wenn du mehrere Knochen über [shift] + [A] → **Armature** → **Single Bone** einfügst, so haben dies Knochen untereinander keine Beziehung. Sie gehören auch verschiedenen Armaturs an (s. Outliner).
- Du kannst aber auch einen einzelnen Knochen mit [shift] + [D] duplizieren. Als Folge entsteht nicht nur eine neuer Knochen, sondern eine ganz neue Armature, wie du im Outliner leicht erkennen kannst.

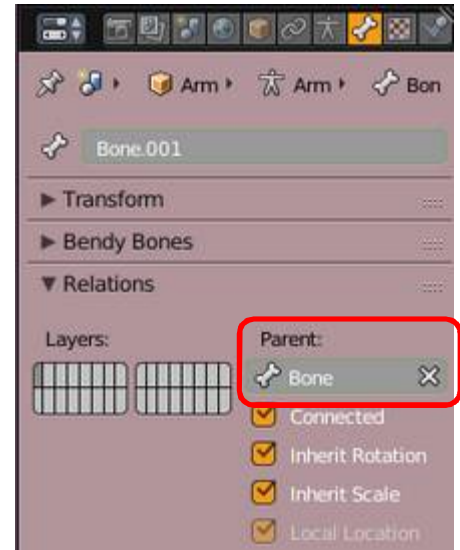
### Edit Mode

Bei Wirbeltieren hängen oft mehrere Knochen in einer Reihe aneinander. Denke an die Wirbelsäule oder Arme und Beine. Aber auch bei vielen Maschinen sind bewegliche Teile in Reihen bzw. Ketten angeordnet. Am häufigsten wirst du also Knochenketten erstellen.

- Eine Knochenkette kannst du durch Extrusion erstellen:
  - Wähle im Object Mode eine Armature (bzw. einen Bone) aus.
  - Wechsle in den Edit Mode und selektiere die Spitze des Knochens (schmale Seite), an den du einen weiteren Knochen „anhängen“ willst.
  - Extrudiere von dort aus einen weiteren Knochen mit [E] und Ziehen mit LMT.
  - Diesen Vorgang kannst du beliebig oft wiederholen und auf diese Weise eine Knochenkette aufbauen.
- Du kannst (wie bei Meshes) einen Knochen unterteilen:
  - Selektiere einen Bone.
  - Im Toolbar (links) findest du unter **Tools** → **Armature Tools** → **Modeling** die Schaltfläche **Subdivide**. Nach einem Klick darauf wird der Bone zweigeteilt.
  - Auch auf diese Weise entsteht eine Knochenkette.



- Die einzelnen Knochen dieser Knochenkette stehen alle in einem Parent-Child-Verhältnis zueinander:
  - Die einzelnen Knochen der Kette werden von Blender automatisch benannt: Der Ausgangsknochen heißt Bone, der daraus extrudierte Knochen Bone.001, der daraus extrudierte Knochen Bone.002 usw.
  - In den Properties → Bone kannst du diese Namen sehen (und verändern):
  - In dem darunter liegenden Panel Relations ist außerdem angezeigt, dass der Parent von Bone.001 der Ausgangsknochen Bone ist.
  - Die Konsequenz ist (wie bei allen Parent-Child-Relationen), dass der Parent-Bone die Children-Bones mitbewegt, die Bewegung eines Childs aber den Parent unbewegt lässt.

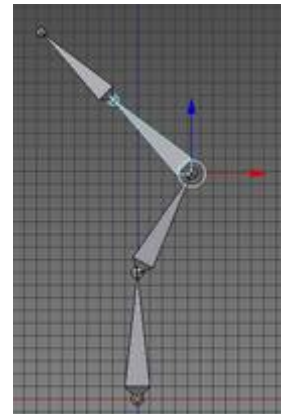


### 7.1.5. Pose Mode

Du kannst die Bones im Edit-Modus einzeln bewegen. Zur Bewegung des gesamten Rigs und einzelner Knochen verfügt Blender aber über einen eigenen Modus, den **Pose Mode**.

Nur im Pose Mode kann ein Skelett animiert und in seiner Funktion getestet werden.

- Selektiere in der Viererkette von Bones den zweiten Bone (Bone.001). Du siehst: Bei Selektion im Pose Mode bekommen die Bones einen blauen Rand.
- Bones werden in der Regel rotiert. Drücke daher [R] und bewege die Maus: **Bone** bleibt unbewegt, **Bone.001** rotiert in seiner Wurzel und nimmt die anderen Bones mit.
- Wenn du jetzt **Bone.002** selektierst und rotierst, entsteht dieses Ergebnis:



#### Tip:

Wenn du die Ausgangslage der Knochen wieder herstellen willst, kannst du dies (im Pose Mode) erreichen durch:

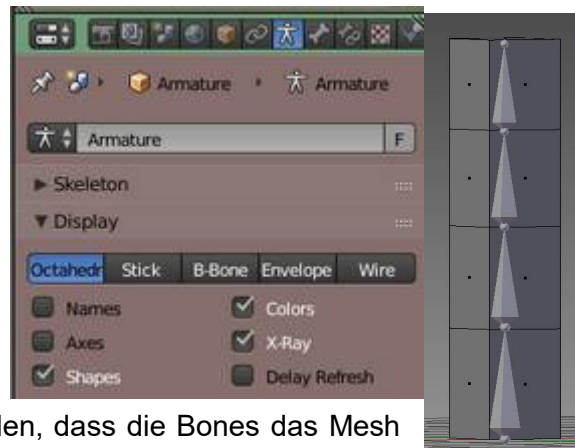
[A] (Selektieren der gesamten Armatur)

[alt] + [R]

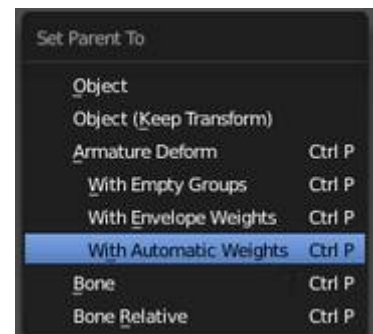
[alt] + [G]

## Beispiel 1: Verbindung der Armature mit einem Mesh

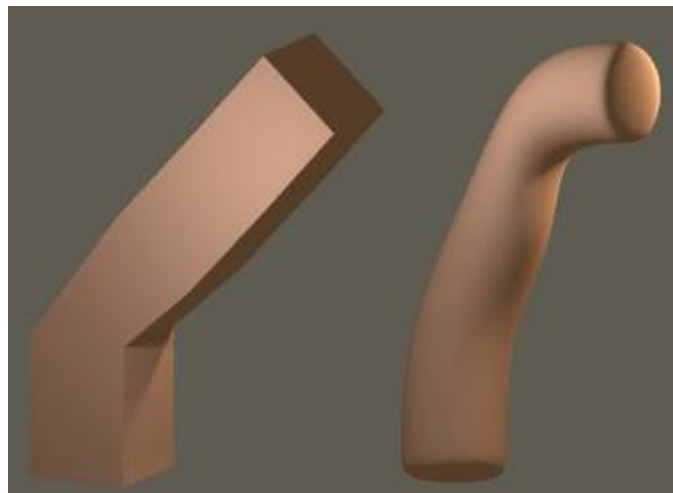
- **Selektiere die *Armature* und aktiviere in den Properties → **Object Data** zur Armature → Unterpanel **Display** die Funktion **X-Ray**. So bleiben bei den folgenden Schritten die Bones sichtbar.**
- **Füge im Nullpunkt einen Würfel ein und skaliere ihn so, dass der untereste Bone von dem Würfel umschlossen wird.**
- **Extrudiere den Würfel dreimal nach oben, so dass dieses Gebilde (rechts)entsteht:**
- **Jetzt muss dieses Mesh (aus den einzelnen Würfeln) mit der Armature so verbunden werden, dass die Bones das Mesh bei Bewegungen quasi mitnehmen.**
  - **Wechsle in den Object Mode und selektiere das Mesh mit RMT.**
  - **Selektiere *danach* (Reihenfolge!) die Armature.**
  - **Drücke [strg] + [P] → **Set Parent To** → **Armature Deform** → **With Automatic Weights**.**



7



- **Wechsle in den Pose Mode, selektiere Bone.001 und rotiere mit [R] und Ziehen. Das Resultat (zur besseren Sichtbarkeit etwas eingefärbt):**
- **Daneben dasselbe Mesh, aber mit Subdivision Surface (Stufe 3) abgerundet. Das könnte schon fast Teil eines organischen Gebildes sein.**



### Tip:

Wenn bei der Bewegung eines Bone das zugehörige Mesh „ausfranst“, liegt dies daran, dass das Mesh versehentlich auch noch mit einem anderen Bone verbunden ist. Sobald du im **Weight Paint** Mode diese Verbindung löschst, ist der Fehler beseitigt.



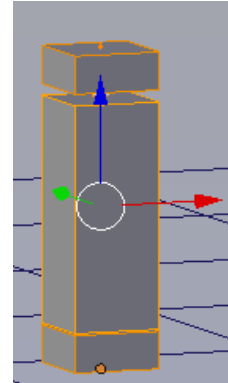


## 7.1.6. Bendy Bones

Bei einem gebogenen Mesh (z.B. eine Schlange, ein Wurm o.ä.) könnte man dessen Form mit mehreren kleinen Bones nachbilden. Einfacher ist es, mit Bendy Bones zu arbeiten. Damit kann ein einzelner Bone in mehrere Abschnitte zerlegt werden, es bleibt aber *ein* Bone.

### Beispiel 2

- Erzeuge einen Bone; skaliere ihn größer und wähle die Darstellungsform B-Bone (s.o.)
- Wechsle in den Edit Mode und unterteile den Bone mit **Subdivide** in drei Teile. Skaliere den oberen und den unteren Bone etwas kleiner. Diese sollen gleich als *Handles* (Anfasser) fungieren (s.u.).



- In der Hierarchie der Bones ist der erste (untere) Bone der *Head* (In-Handle), der mittlere Bone ist der *Bendy Bone*, der oberste ist der *Tail* (Out-Handle). Im *Outliner* wurden die Teile der Armature in dieser Weise umbenannt.



- Selektiere den BendyBone in der Mitte und rufe in den Properties das *Bone*-Menü (Knochen-Icon) auf; trage im Untermenü *Bendy Bones* **Segments** = 10 ein.

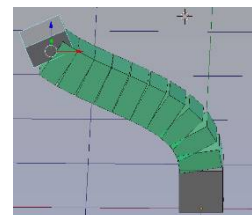


- Im 3D Viewer erkennst du sofort, dass der BendyBone in 10 Segmente unterteilt wurde.
- Wenn du jetzt einmal probierhalber in den Pose Mode wechselst, wirst du feststellen, dass du den oberen Handle zwar rotieren, aber nicht bewegen kannst. Das soll jetzt geändert werden:

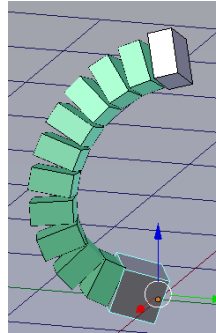
- Selektiere (im Edit Mode) den *Tail* im Outliner.
- Drücke [alt] + [P] → Clear Parent.
- Das Ergebnis:
- Wechsle in den *Pose Mode* und selektiere den *Bendy Bone*.



- Rufe in den Properties das Menü der *Bone Constraints* (Icon mit Knochen und Kettenglied) auf.
- Klicke auf *Add Bone Constraint* → Tracking → **Stretch to**.
- Treg ein: Target = Armature; Bone = Tail
- Wenn du jetzt den Tail selektierst, [G] drückst und die Maus bewegst, kannst du den Bendy Bone beliebig bewegen. Allerdings bewegen sich die Handles bei Rotationen noch nicht mit



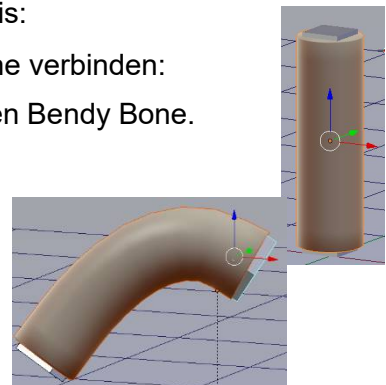
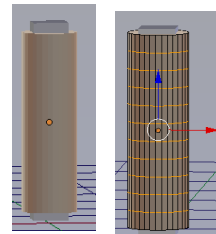
- Gehe noch einmal in das *Bendy Bones*-Menü, setze einen Haken vor **Use Custom Handle References** und trage darunter ein bzw. rufe auf: In = *Head* und Out = *Tail*. Jetzt kannst du die Handles beliebig rotieren. Sie zeigen die gleiche Rotation wie die Teile im Bendy Bone:



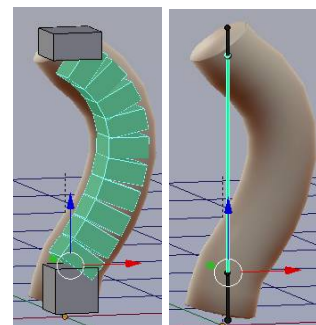
- Selektiere die gesamte Armatur und bringe sie mit [alt] + [G] und [alt] + [R] wieder in die Ausgangslage.

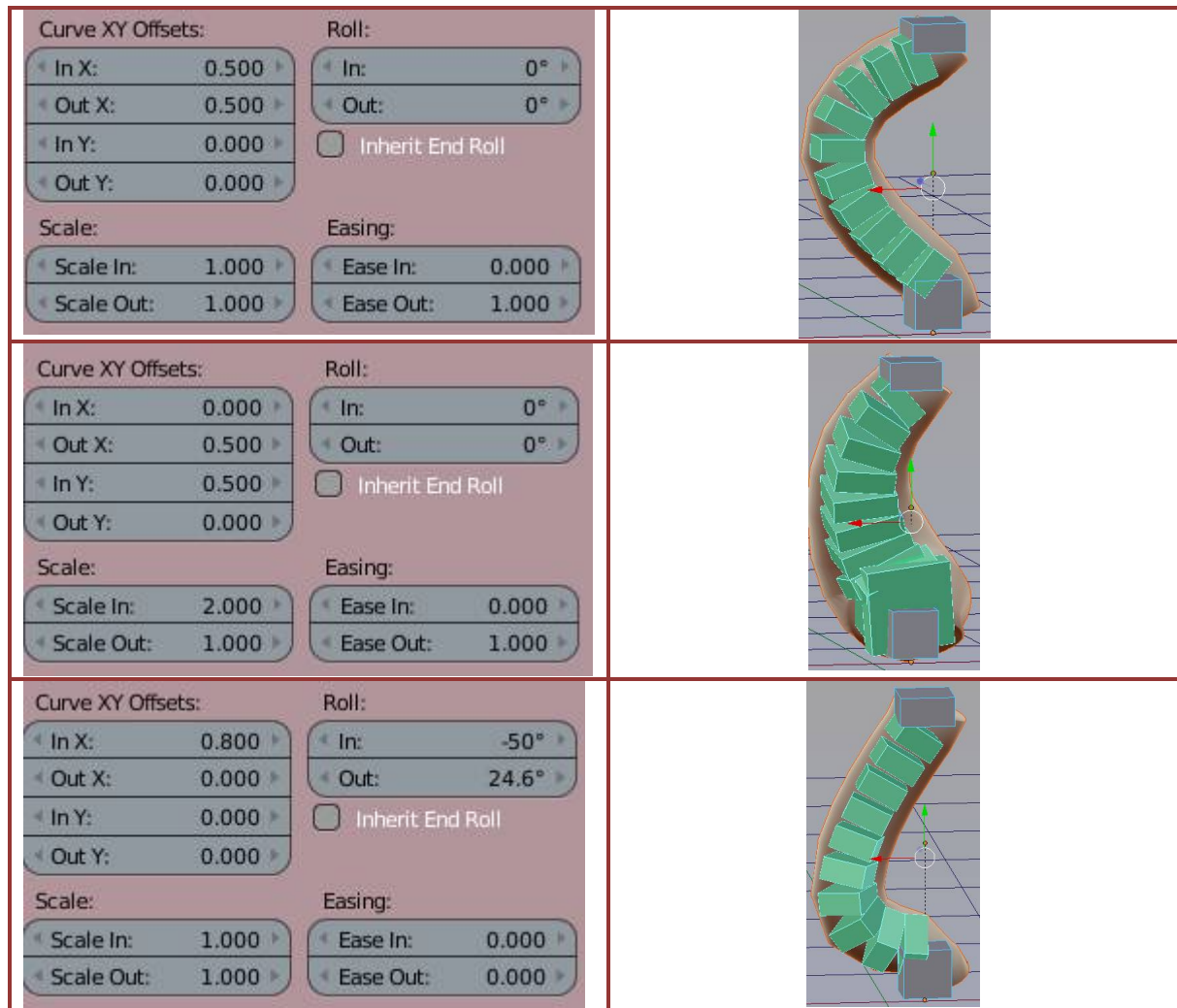


- Wechsle jetzt wieder in den Object Mode und füge als Mesh einen **Cylinder** hinzu. Skaliere und positioniere ihn so, dass er den Bendy Bone umschließt. Die beiden Handles sollten oben und unten zu sehen sein.
- Wechsle in den Edit Mode und erzeuge mit **Loop Cut and Slide** Segmente auf dem Zylinder – entsprechend den Segmenten auf dem Bendy Bone.
- Wenn du die Kanten des Zylinders oben und unten noch etwas extrudierst und nach innen skalierst und dann im Object Mode die Funktion **Smooth** anwendest, erhältst du etwa dieses Ergebnis:
- Als Nächstes musst du den Zylinder mit dem Bendy Bone verbinden:
  - Selektiere im Object Mode den Zylinder **und dann** den Bendy Bone.
  - Drücke [strg] + [P] → With Automatic Weights
- Wenn du jetzt die Handles im Pose Mode selektierst, kannst du den Zylinder beliebig verformen.
- Es ist leicht vorstellbar, dass auf diese Weise z.B. die Fortbewegung eines Wurms dargestellt werden kann.
- Schließlich noch einige Details:



- Die charakteristische Ansicht von Bendy Bones erscheint nur dann, wenn zuvor *B-Bone* aktiviert wurde. Wurde eine andere Ansicht aktiviert (z.B. *Stick*), so wird der Bendy Bone einfach als gerader Knochen dargestellt. Gleichwohl behält er seine Funktion (Verformung von Meshes) bei.
- Im Untermenü *Bendy Bones* können verschiedene Parameter eingetragen werden, die alle in **In** und **Out** unterschieden sind. Erinnerung an die Eintragungen zu *Use Custom Handle References*. Dort haben wir festgelegt: In = *Head* und Out = *Tail*. Darauf beziehen sich diese Hinweise. Wenn du einen *In*-Parameter veränderst, geht die „Initiative“ gewissermaßen vom *Head* (unten) aus, wenn du einen *Out*-Parameter veränderst, vom *Tail* (oben).





Die einzelnen Parameter können animiert werden, du kannst also z.B. den oberen Teil des Mesh durch Skalieren im Feld **Scale Out** größer und kleiner werden lassen.

Insgesamt wurden die Möglichkeiten zur Animation durch Riggung mit den *Bendy Bone* Funktionen wesentlich erweitert. Weitere Erläuterungen und Anwendungsmöglichkeiten findest du unter:

[https://www.youtube.com/watch?v=BsJ3Grq\\_3GM](https://www.youtube.com/watch?v=BsJ3Grq_3GM)

[https://www.youtube.com/watch?v=JS8V4\\_Ncn0w](https://www.youtube.com/watch?v=JS8V4_Ncn0w)

[https://www.youtube.com/watch?v=WG3PhuzGz\\_o](https://www.youtube.com/watch?v=WG3PhuzGz_o)

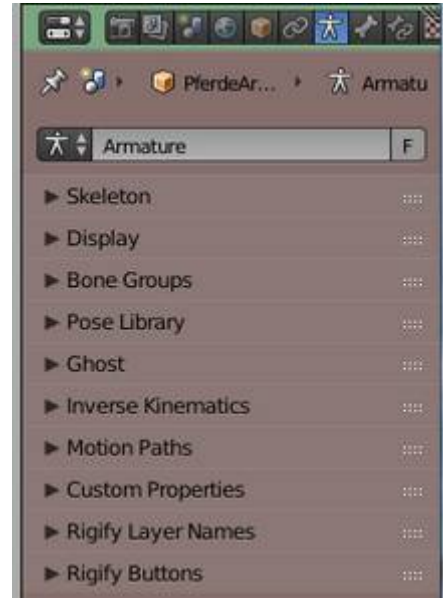
<https://www.youtube.com/watch?v=5zmWmJLfspg>

<https://www.youtube.com/watch?v=2J9uqWkK-w8>

Es liegen noch kaum Erfahrungen über die Anwendung von Bendy Bones beim Rigging von Menschen oder Tieren vor, diese Funktion könnte sich aber revolutionär auswirken und die bisherigen Vorgehensweisen ablösen.

## 7.2 Einstellungen der Armatures

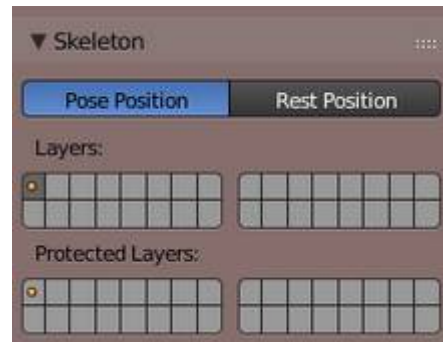
Die *Object Data* der *Armature* können in diesen Panels dargestellt und verändert werden:



11

### Skeleton

- **Pose Position:** In dieser Einstellung kann die Stellung der Bones im **Pose Mode** verändert werden.
- **Rest Position:** Hiermit lässt sich die Armature in die Position zurückversetzen, die du im Edit Mode festgelegt hast. Im Pose Mode kannst du dann die Bones nicht bewegen.
- **Layers:** Einzelne Knochen können (ähnlich wie Objekte) in 32 verschiedenen Layern abgelegt werden. Dies dient vor allem der Übersichtlichkeit bei Armatures mit einer großen Anzahl von Bones. Das funktioniert ähnlich wie bei Objekten:
  - Bone selektieren.
  - [M] drücken.
  - Auf den gewünschten Layer klicken.



**Hinweis:** Das Verschieben von Bones in andere Layer funktioniert nur im Skeleton Panel – nicht im Bone Panel!

### Display

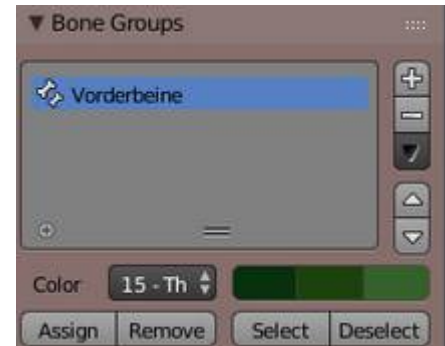
- Zur Bedeutung der Schaltflächen in der obersten Zeile siehe Abschnitt *Darstellungsformen von Bones* im Kapitel 7.1.
- **Names:** Wenn aktiviert, werden neben die Bones deren Namen eingeblendet.
- **Axes:** Wenn aktiviert, wird neben den Bones deren Ausrichtung im Koordinatensystem eingeblendet.
- **Shapes:** Wenn aktiviert, nehmen Bones die Formen der Meshes an, die ihnen als **Custom Shape** zugewiesen wurden (siehe Abschnitt *Darstellungsformen von Bones* im Kapitel 7.1). Wenn nicht aktiviert, werden immer die Originalformen der Bones angezeigt.
- **Colors:** Wenn aktiviert, werden die Bones in den Farben dargestellt, die du ihnen für bestimmte Bone Groups (s. nächster Abschnitt) zugewiesen hast.



## Bone Groups

Hier kannst du Gruppen von Knochen zusammenfassen und ihnen eigene Farben zuweisen:

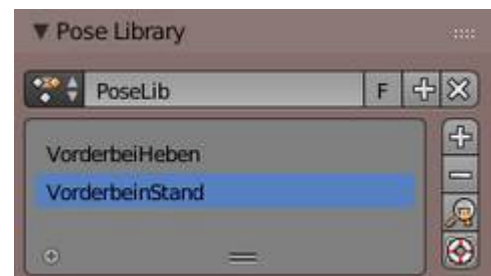
- Klick auf das Plus-Zeichen. Es erscheint das Wort **Groups** im Anzeigefeld. Doppelklick darauf und du kannst den Namen überschreiben – hier in *Vorderbeine* (von einem Pferd).
- Klick auf das Feld **Color Set** und wähle aus dem Pop-Up-Menü eine geeignete Farbkombination aus.
- Selektiere die Bones, die der Gruppe angehören sollen.
- Klick auf **Assign**.



## Pose Library

Dies ist eine sehr arbeitssparende Einrichtung: Bestimmte Posen (Einstellung von Bones) können hier abgespeichert und später wieder aufgerufen werden.

- Im oberen Feld, in dem standardmäßig PoseLib eingetragen ist, können verschiedene Bibliotheken benannt und aufgerufen werden. Klick auf das Pluszeichen in dieser Zeile bewirkt den Aufruf einer neuen Bibliothek. Klick auf das Feld ganz links in dieser Zeile öffnet eine Auswahl der bisher eingerichteten Bibliotheken.
- In dem Feld darunter sind die zu einer Bibliothek gehörenden Posen aufgelistet:
  - Bringe die Knochen zunächst in die gewünschte Stellung.
  - Klick auf das Pluszeichen richtet dann die neue Pose ein. Dabei gibt es drei Möglichkeiten:
    - Add New: Fügt der aktiven Bibliothek die Pose neu hinzu.
    - Add New (Current Frame): Fügt die neue Pose an der Stelle in der Timeline hinzu, wo sich der Cursor aktuell befindet. (weniger empfehlenswert)
    - Replace Existing: Ersetzt die gerade selektierte Pose durch die neue.
- Klick auf das Minuszeichen beseitigt die selektierte Pose aus der Bibliothek.
- Klick auf die Lupe aktiviert die in der Liste ausgewählte (blau hinterlegte) Pose.
- Klick auf den Rettungsring startet eine Rettungsaktion: Hierdurch wird die Aktionen in den Keyframes der Timeline in die Bibliothek übernommen.



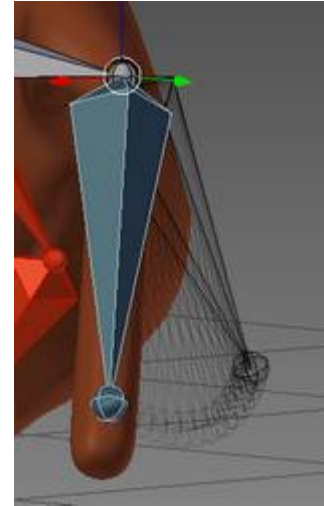


## Ghost

Mit dieser Funktion können bewegte Knochen über mehrere Frames hinweg angezeigt werden. (Etwas Ähnliches gibt es beim *Grease Pencil* mit der *Onion Skin*-Funktion.)

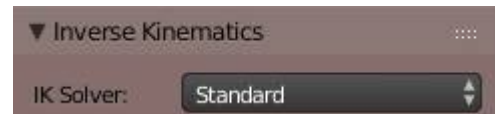


- **Around Frame:** Die Knochenstellungen werden vor (links) und nach (rechts) dem aktuellen Frame angezeigt.
- **In Range:** Die Knochenstellungen werden vor (links) und nach (rechts) dem aktuellen Frame nur in dem angegebenen Bereich (Range) angezeigt.
- **On Keyframes:** Die Knochenstellungen werden vor (links) und nach (rechts) dem aktuellen Frame nur für Keyframes angezeigt.
- **Selected Only:** Bei Aktivierung werden nur die Knochenstellungen für den (die) selektierten Knochen gezeigt.
- **Range:** Anzahl der Frames vor und nach dem aktuellen Frame, für die die Knochenstellungen gezeigt werden.
- **Step:** Die Zahl gibt an, der wievielte Frame vor und nach dem aktuellen Frame gezeigt werden soll.
- **Start / End** (nur bei **In Range**): Gibt die Nr. der Frames an, bei denen die Anzeige anfangen und beendet werden soll.



## Inverse Kinematics

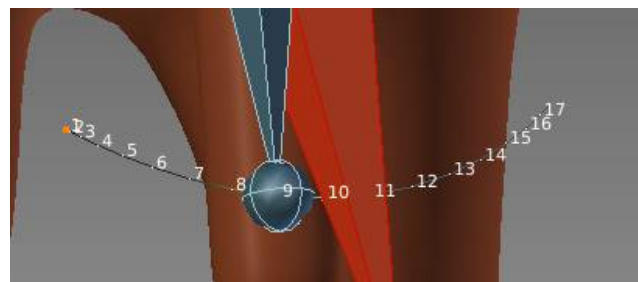
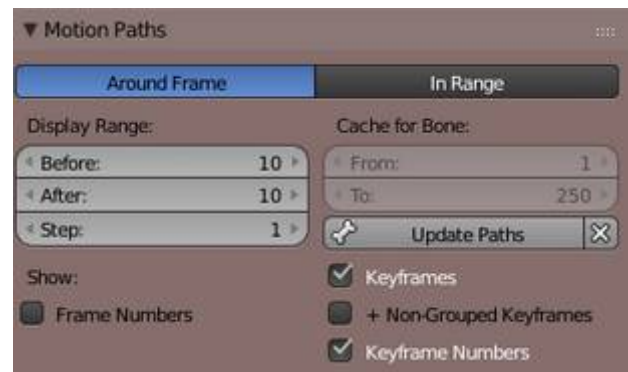
Hier kannst du auswählen, nach welcher Methode **Inverse Kinematics** berechnet werden sollen. (s. Kap. 7.4.4)



## Motion Paths

Wie bei der gleichlautenden Funktion für Meshes wird hier die Bewegungslinie eines Knochens nachgezeichnet.

- Die Berechnung beginnt, wenn du auf die Schaltfläche **Calculate** klickst. Danach ändert diese ihre Bezeichnung in **Update Parts**.
- **Around Frame** bzw. **In Range:** Analoge Wirkungsweise wie bei Ghost.
- Die übrigen Funktionen erklären sich von selbst.



## 7.3 Einstellungen der Bones

Es stehen (**im Pose Mode**) diese Panels zur Verfügung:

Einstellungen der Bones betreffen immer den gerade selektierten Bone. In dem Beispiel ist dies der Knochen mit der Bezeichnung *Bone.001*.



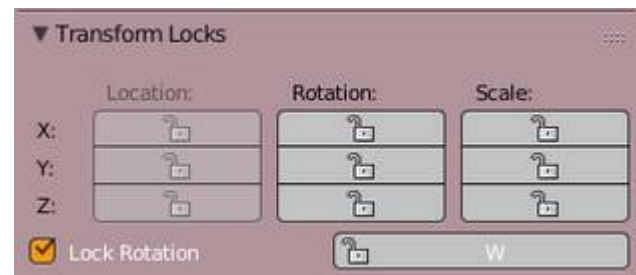
### Transform

Hier werden dieselben Felder angezeigt wie im Transform-Panel der Objects. In der Regel benötigst du nur die Rotationen.

Wenn du in den Feldern der Spalte *Scale* Veränderungen vornimmst, ändert sich nicht nur die Größe des Bones und des zugeordneten Mesh', sondern auch die aller Children dieses Bones und die zugehörigen Teile des Mesh.

### Transform Locks

Auch diese Einstellungen sind aus dem Transform Panel für Objects bekannt: Durch Klick auf das einzelne Schloss kannst du Transformations-, Rotations- oder Skalierungsmöglichkeiten blockieren.

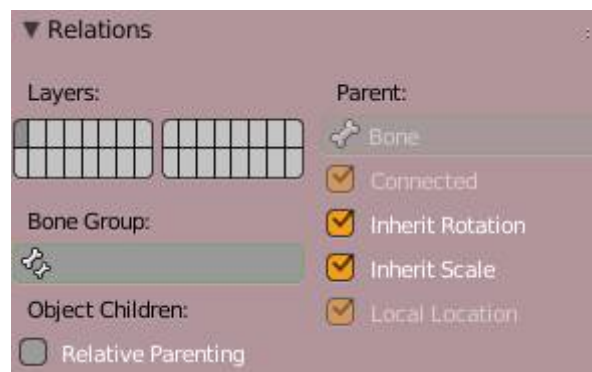


### Relations

**Layers:** Wenn du Wie schon im Abschnitt *Mehrere Knochen und Knochenketten* von Kap. 7.1 erwähnt, kannst du einzelne Knochen in einen der 32 Layer verschieben. (s. Abschnitt *Skeleton* in Kap. 7.2) Wenn du einen Bone selektierst, wird dir hier angezeigt, in welchen Layer er gehört.

**Parent:** Zeigt an, welcher Bone der Parent zu diesem Knochen ist. Du kannst an dieser Stelle den gewünschten Parent auch eintragen.

**Bone Group:** Zeigt an, zu welcher Bone Group dieser Knochen gehört. Du kannst an dieser Stelle die gewünschte Bone Group auch eintragen.



Stelle die gewünschte Bone Group auch eintragen.

**Connected:** Wenn Häkchen, wird der Child Bone mit dem Parent Bone verschmolzen.

**Inherit Rotation:** Wenn Häkchen, wird die Rotation des Parent Bone auf den Child Bone übertragen.

**Inherit Scale:** Wenn Häkchen, wird die Skalierung des Parent Bone auf den Child Bone übertragen.

**Local Location:** Wenn Häkchen, wird das lokale Koordinatensystem des Knochens verwendet.

**Relative Parenting:** Wenn Häkchen, (im Falle einer Parent-Child-Beziehung) wird die Parent Operation relativ ausgeführt.

## Display

Siehe Abschnitt *Darstellungsformen von Bones* in Kap. 7.1.3.

**Hide:** Wenn Häkchen, wird der Knochen nicht gezeigt.



## Inverse Kinematics (IK)

Genauere Erläuterungen zum Thema s. Kap. 7.4.4, Abschnitt *Bewegungsmöglichkeiten einschränken*.

## Deform

**Deform:** Wenn Häkchen, verformt der Knochen das Mesh.

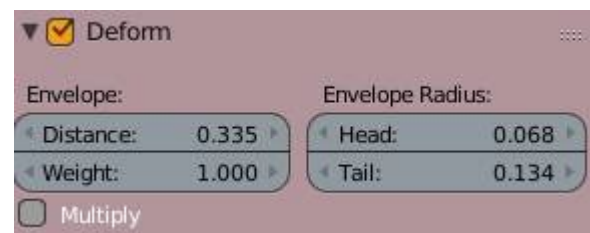
**Envelope:** Die darunter stehenden Einstellungen gelten für Knochen vom Typ **Envelope**.

**Distance:** Größe des Einflussbereichs des Bones.

**Weight:** Gewicht, mit dem die Vertices mit dem Bone innerhalb des Einflussbereichs verbunden sind.


**Multiply:** Wenn Häkchen, werden die Vertices-Werte mit dem Einflussbereich multipliziert.

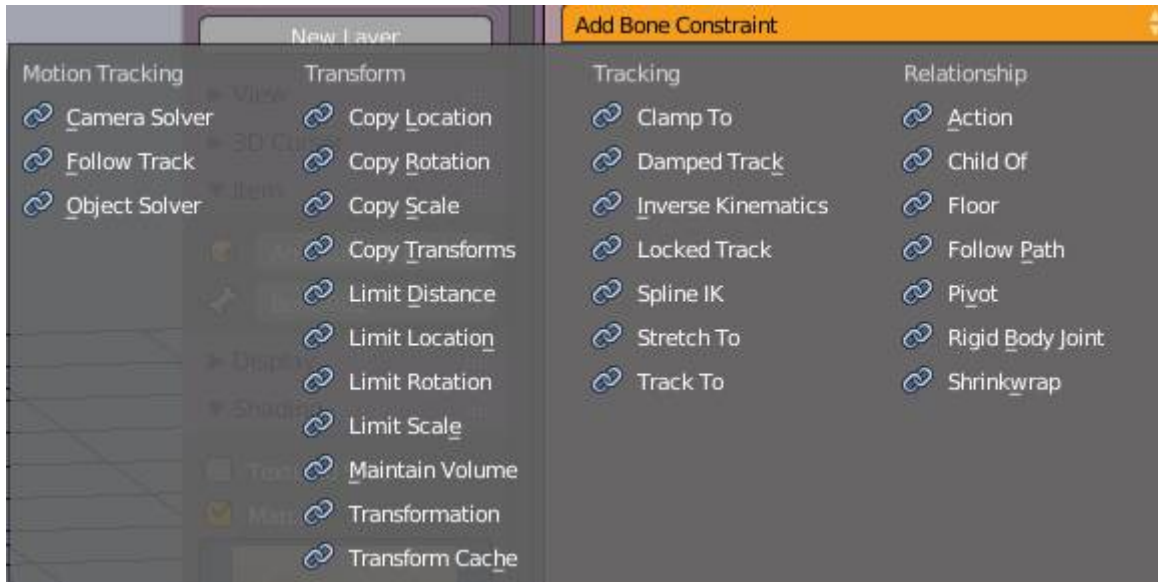
**Radius** – gilt nur für Knochen vom Typ Envelope: Größe von Head und Tail des Bones und damit auch dessen Einflussbereich.





## 7.4 Bone Constraints

- **Bone Constraints** beziehen sich nur auf Bones; sie werden durch Klick auf das Icon mit Knochen und Kettenglied  aufgerufen und können nur im Pose Mode gesetzt werden.
- Wenn du auf die Schaltfläche **Add Bone Constraint** klickst, öffnet sich diese Auswahl:
- In den Spalten sind die vier Typen von Bone Constraints angeordnet: **Motion Tracking**, **Transform**, **Tracking** und **Relationship**.



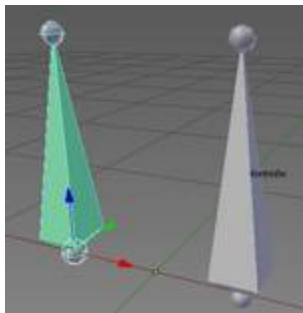
Wenn du Kap. 6.5 betrachtest, wirst du feststellen, dass dort die gleiche Aufstellung für *Object Constraints* gilt.

### 7.4.1 Transform Bone Constraints

Die ersten vier Constraints sind Copy-Constraints. Sie bewirken, dass der betr. Knochen alles nachmacht – eben kopiert, was der Kontrollknochen tut. Du brauchst also immer einen zweiten Knochen, der den anderen steuert. Dabei wird die Constraint-Eigenschaft immer dem zu steuernden Knochen zugewiesen.

#### Copy Location

- Nehmen wir an, du hast deiner Szene zwei Bones eingefügt. Der linke Bone (grün) sei der zu steuernde Bone (Owner); er gehört zur Armature **Armature**. Der rechte Bone



trägt den Namen **Kontrolle** und gehört zur Armature **Steuerung**. Wenn du nun den linken Bone selektierst und dann Constraint **Copy Location** anwählst, erhältst du nebenstehendes Panel zum ausfüllen:



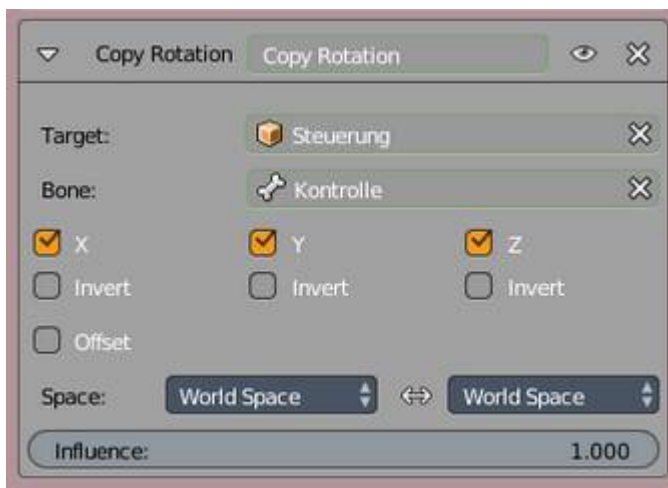
- **Target:** In das Feld ist der Name der Armature einzugeben, die den Kontroll-Bone enthält. In unserem Fall lautet dieser Name *Steuerung*.
- **Bone:** In das Feld ist der Name des Kontroll-Bones einzugeben. In unserem Fall lautet

dieser Name *Kontrolle*.

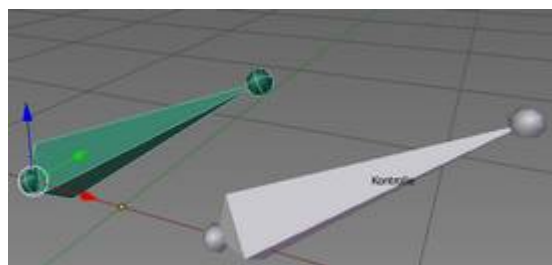
- **Head/Tail:** Mit dem Schieberegler kannst du festlegen, welcher Teil des Kontroll-Bones für die Steuerung maßgeblich ist – Head oder Tail.
- **X, Y, Z:** Mit dem Haken legst du fest, in welche Richtung der betr. Achse die Verschiebung erfolgen soll.
- **Invert:** Wenn Haken gesetzt, erfolgt die Verschiebung in die entgegengesetzte Richtung zu der des Kontroll-Bones.
- **Offset:** Wenn du den Bone schon vorher (z.B. im Object Mode) verschoben hast, erfolgt die Verschiebung durch den Kontroll-Bone relativ zu diesem Standort, wenn der Haken gesetzt ist.
- **Space:** Standard ist das Welt-Koordinatensystem. Du kannst aber auch noch drei andere verwenden: **Local Space**, **Local with Parent** und **Pose Space**.
- **Influence:** Mit dem Schieberegler kannst du festlegen, wie stark sich die Bewegungen des Kontroll-Bones auswirken sollen. Bei Influence = 0.50 bewegt sich z.B. der grüne Bone ein halbe Einheit, wenn der Kontroll-Bone und eine ganze Einheit bewegt wird.

## Copy Rotation und Copy Scale

- Das zugehörige Panel ist genauso aufgebaut wie bei Copy Location, allerdings ohne die Zeile Head/Tail. Wenn du jetzt den Kontroll-Bone drehst, dreht sich der grüne Bone mit.

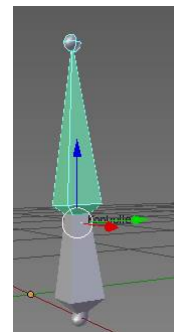


- Copy Scale funktioniert nach demselben Prinzip: Wenn du den Kontroll-Bone skalierst, ändert sich auch die Größe des grünen Bone. Eine Invert-Funktion gibt es hier allerdings nicht.



## Copy Transforms

- Das Prinzip ist dasselbe, aber die Wirkung eine andere:
  - Der grüne Bone macht alle Transformationen mit, die der Kontroll-Knochen macht: Location, Rotation und Skalierung.
  - Wenn Head/Tail = 0.00, fallen beide Bones ineinander. Wenn Head/Tail = 1.0, schiebt sich der grüne Bone an die Spitze des Kontroll-Bone. In dem Beispiel ist der Wert 0.5 eingestellt.



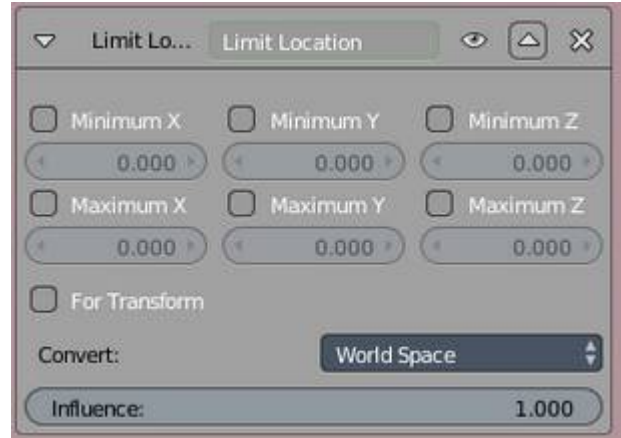
Die Funktion der folgenden vier Constraints bezieht sich auf Einschränkungen – und das passt dann auch zu der Bezeichnung:

## Limit Distance

- Dieser Constraint bewirkt, dass der grüne Bone eine bestimmte Distanz zum Kontroll-Bone nicht überschreitet. Beide Bones können näher aneinander-rücken, sich aber nicht weiter voneinander entfernen.

## Limit Location / Rotation / Scale

- Alle drei Constraints bewirken, dass der Bone nur innerhalb bestimmter Grenzen transformiert werden kann. Ein Kontroll-Bone ist dabei überflüssig.
  - Die Grenzen sind jeweils im Hinblick auf die drei Achsen X, Y und Z anzugeben.
  - Bei *Limit Rotation* werden die Grenzen in Grad angegeben.



## Maintain Volume

Wie der Name sagt, soll das Volumen des gesteuerten Bone bzw. des zugehörigen Mesh' erhalten.

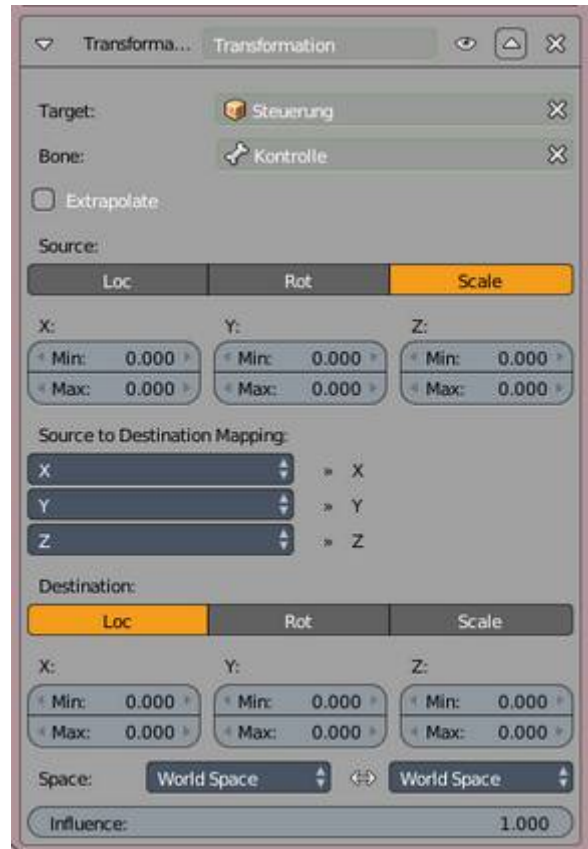
In der Zeile *Free* kannst du anklicken, welche Dimension bei der Skalierung verändert werden kann: X, Y oder Z.



## Transformation

Der *Transformation Constraint* kombiniert die Copy Loc/Rot/Scale constraints. Es ermöglicht damit, verschiedene Bewegungsarten in andere zu überführen, sie eben zu transformieren.

Das Prinzip ist der folgenden Tabelle zu entnehmen. Dabei läuft der Einfluss immer von der *Source* zur *Destination*.



|  |       | Daten der Source (des steuernden Bone)  |                                  |                                    |
|--|-------|---|----------------------------------|------------------------------------|
|  |       | Loc                                     | Rot                              | Scale                              |
| Daten der Destination (des gesteuerten Bone) | Loc   | Ortsveränderung erzeugt Ortsveränderung | Rotation erzeugt Ortsveränderung | Größenänderung erzeugt Ortsveränd. |
|  | Rot   | Ortsveränderung erzeugt Rotation        | Rotation erzeugt Rotation        | Größenänderung erzeugt Rotation    |
|  | Scale | Ortsveränderung erzeugt Größenänderung  | Rotation erzeugt Größenänderung  | Größenänderung erzeugt Größenänd.  |

### Variante 1

- Selektiere *Bone* und rufe *Transformation Constraint* auf.
- Trage ein:
  - Target = Steuerung
  - Bone = Kontrolle
  - Source = Loc
  - X-Max = 1.0
  - Source to Destination Mapping = alle X
  - Destination = Loc
  - X-Max = 1.0
- Wenn du jetzt den Kontroll-Bone selektierst und z.B. mit [G] in der X-Richtung bewegst, folgt der (gesteuerte) Bone um eine Blendereinheit – aber nur in der X-Richtung. Die Bewegung des Kontroll-Bone in andere Richtungen und über eine Blendereinheit hinaus hat keine Auswirkungen.

### Variante 2

---

- Alle Einstellungen wie eben mit dem Unterschied:
  - Unter Destination = Loc ist X-Max = 0.0, aber Z-Max = 1.0
- Wenn du jetzt den Kontroll-Bone selektierst und z.B. mit [G] in der X-Richtung bewegst, folgt der (gesteuerte) Bone um eine Blendereinheit – aber nur in der Z-Richtung, also nach oben. – Eine horizontale Bewegung (der Steuerung) wurde also in eine vertikale Bewegung umgewandelt (transformiert).

### Variante 3

---

- Alle Einstellungen wie eben mit dem Unterschied:
  - Destination = Rot
  - X-Max = 0.0, aber Z-Max = 0.0, aber Y-Max = 360°.
- Wenn du jetzt den Kontroll-Bone selektierst und z.B. mit [G] in der X-Richtung um eine Blendereinheit bewegst, rotiert der (gesteuerte) Bone – um 360° um die Y-Achse. – Eine horizontale Bewegung (der Steuerung) wurde also in eine Rotations-Bewegung umgewandelt (transformiert).

### Variante 4

---

- Alle Einstellungen wie eben mit dem Unterschied:
  - Destination = Scale
  - X-Max = 0.0, Y-Max = 1.0, Z-Max = 1.0.
- Wenn du jetzt den Kontroll-Bone selektierst und z.B. mit [G] in der X-Richtung um eine Blendereinheit nach links bewegst, verschwindet der (gesteuerte) Bone, wenn du ihn nach rechts bewegst, hat der Bone wieder seine volle Größe. – Eine horizontale Bewegung (der Steuerung) wurde also in eine Skalierungs-Bewegung umgewandelt (transformiert).

Sehr instruktive Tutorials findest du unter

<http://www.bing.com/videos/search?q=blender+bone+constraints+transformation&view=detail&mid=A23F77BA1670979248C5A23F77BA1670979248C5&FORM=VIRE>

und

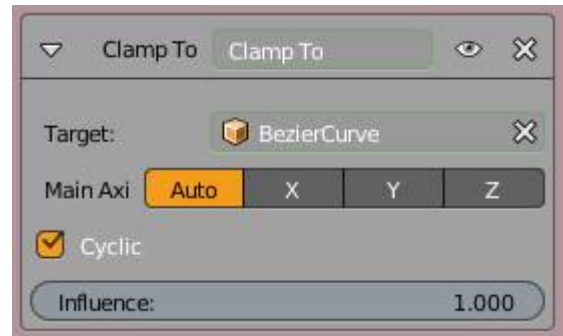
<http://www.bing.com/videos/search?q=blender+bone+constraints+transformation&&view=detail&mid=D552AEF5D56FFBD5DFC7D552AEF5D56FFBD5DFC7&FORM=VRDGAR>

## 7.4.2 Tracking Bone Constraints

### Clamp To

Der **Clamp To** Constraint als *Bone Constraint* funktioniert genauso wie sein Bruder von den *Object Constraints* – nur eben mit Bones (vgl. Kap. 6.5.2). Er verbindet einen Bone mit einer Kurve.

- Füge zum Startwürfel eine **BezierCurve** hinzu. [Shift] + [A] → Curve → Bezier. Vergrößere der Kurve etwas.
- Selektiere den Bone.
- Rufe den **Clamp To** Constraint auf und setze als Target die gerade eingefügte **BezierCurve** ein.
- Wenn du den Bone bewegst, siehst du, dass er der **BezierCurve** folgt.
  - Setze den Bone in Frame 1 an den Anfang der Kurve und drücke [I] → LocRot.
  - Setze den Bone in Frame 100 ans Ende der Kurve und drücke [I] → LocRot.
  - Starte die Animation mit [alt] + [A]: Der Bone folgt der Kurve.



### Damped Track

Damped Track bewirkt, dass ein Bone (Owner) einem anderen Objekt (Target) immer zugewandt ist, egal wohin sich die beiden Objekte bewegen. Das zweite Objekt kann ein Mesh oder auch ein Bone sein. (vgl. Kap. 7.5.2)

### Inverse Kinematics

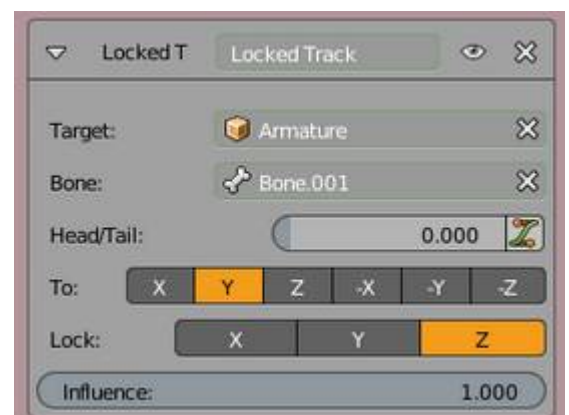
s. Kap. 7.4.4

### Locked Track

Lock Tracked bewirkt, dass der Owner dem Target immer dieselbe Seite zuwendet; dabei kannst du eine der Drehachsen sperren. (vgl. Kap.6.5.2)

Demo:

<https://vimeo.com/171280773>

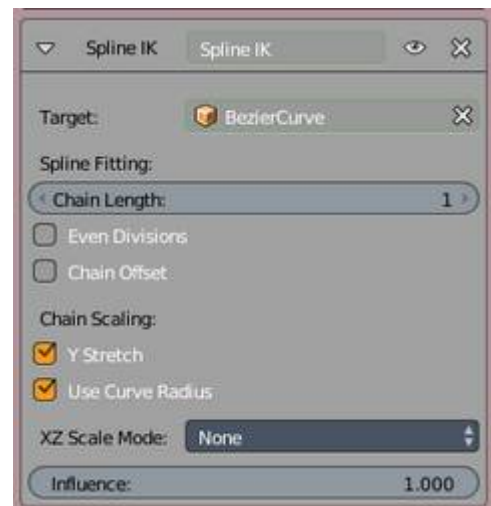




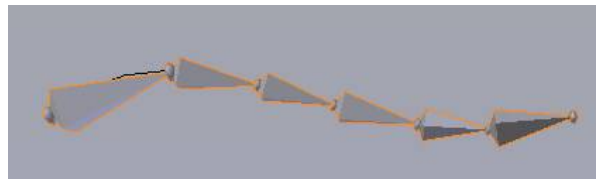
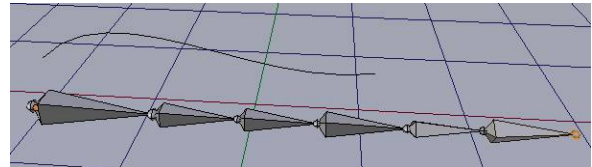
## Spline IK

Das *Spline IK* Constraint richtet eine Knochenkette entlang einer Kurve aus. Es ist ein unschätzbare Werkzeug beim Rigging. Insbesondere ist es geeignet für das Rigging flexibler Körperteile wie Schwänze, Tentakel oder Wirbelsäulen, aber auch für nicht-organische Gegenstände wie Seile.

Um mit Spline IK arbeiten zu können, brauchst du also eine Kette miteinander verbundener Knochen und eine Kurve, an die diese Kette angepasst werden soll:



- Selektiere (im Pose-Mode), den letzten Knochen der Kette.
- Rufe Spline IK auf und gib als Target die *BezierCurve* ein. Der Knochen springt zu der *BezierCurve*.
- Erhöhe jetzt die **Chain Length**. Bei jeder Werterhöhung springt ein weiterer Knochen an die *BezierCurve*.
- Die übrigen Angaben bedeuten:
  - **Even Divisions**: Wenn aktiviert (Häkchen), wird die Länge der Bones bei der Anpassung an die Kurve ignoriert. D.h. die Abschnitte sind alle gleich groß.
  - **Chain Offset**: Wenn aktiviert (Häkchen), wird die Knochenkette nicht mehr an die Kurve angepasst, sondern wieder an die Ausgangsposition verschoben.
  - **Y-Stretch**: Wenn aktiviert (Häkchen), ist die Knochenkette bzgl. ihrer Länge an die Kurve angepasst.
  - **Use Curve Radius**: Wenn aktiviert (Häkchen), wird der durchschnittliche Radius der Endpunkte verwendet, um die Skalierung der Bones in X- und Z-Richtung festzulegen.
  - **XZ Scale Mode**: Skalierung, der ein Bone zur Erreichung seines Target (hier also der Kurve) unterliegt.
    - **None**: Keine Skalierung X- und Z-Achse.
    - **Bone Original**: Original-Skalierung des Bones.
    - **Inverse Scale**: Die Skalierung der X- und Z-Achse ist die Inversion der X-Skalierung.
    - **Volume Preservation**: Bei Veränderung der Skalierung wird das Volumen der Knochen erhalten.
  - **Influence**: Einfluss der Funktion.



<https://vimeo.com/171282278>

## Stretch To

Dieses Constraint bewirkt, dass ein Bone (Owner) sich durch Rotation und Skalierung einem anderen Objekt (Target) annähert. Das ist ein ähnliches Verhalten wie bei Track to (s.u.), Allerdings wird hier angenommen, dass die Y-Achse die Achse ist, die sich streckt, und du kannst keine andere Achse auswählen. Das Target muss kein Bone sein (vgl. Kap. 6.5.2). In der folgenden Beschreibung wird aber davon ausgegangen, dass Owner und Target beides Bones sind.

**Target:** Armatur des Target.

**Bone:** Bone des Target.

**Head/Tail:** Ausmaß, in dem sich der Owner am Head oder am Tail des Target ausrichten soll.

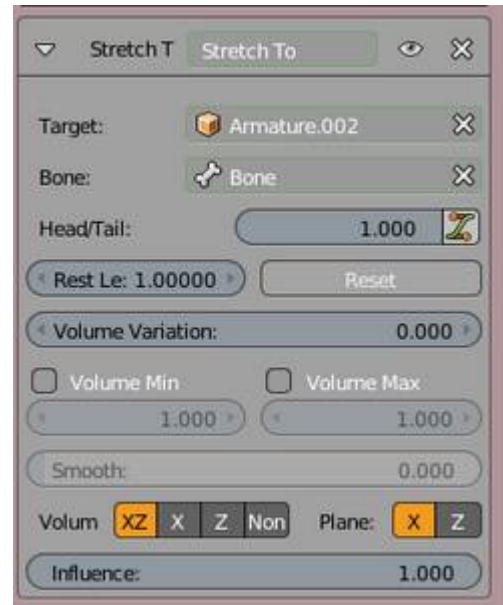
**Rest Length:** Bei 1.0 verbindet sich der Owner direkt mit dem Target; bei Werten  $>1.00$  reicht die Streckung nicht bis zum Target; bei Werten  $<1.00$  geht sie über das Target hinaus.

**Reset:** Bei Klick wird der Wert für die Rest Length neu berechnet, so dass sie dem aktuellen Abstand zwischen Owner und Target entspricht (d.h. dem Abstand vor Anwendung des Constraints).

**Volume Variation:** Der hier eingetragene Wert wirkt sich auf das Ausmaß der Volumen-Variation bei Streckung bzw. Stauchung des Bone aus.

**Volume Min/Max:** Hier kannst du das Ausmaß der zulässigen Volumenänderung festlegen.

**Volume:** Zulässige Veränderungen in den Richtungen der X- bzw. Z-Achse.



<https://vimeo.com/171283118>

## Track To

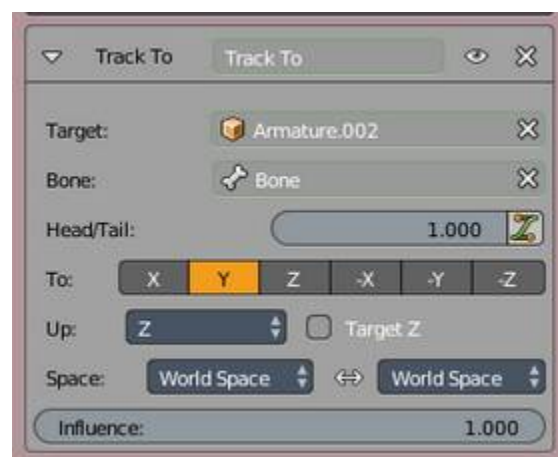
Track To bewirkt, dass der Owner dem Target immer dieselbe Seite zuwendet (vgl. Kap. 6.5.2).

**Target:** Armatur des Target.

**Bone:** Bone des Target.

**Head/Tail:** Ausmaß, in dem sich der Owner am Head oder am Tail des Target ausrichten soll.

<https://vimeo.com/171283522>





### 7.4.3 Relationship Bone Constraints

Die unter dieser Überschrift zu Verfügung stehenden Constraints wurden zu einem großen Teil schon in Kap. 6.5.3 besprochen. Ausnahmen sind der *Action Constraint* und der *Rigid Body Joint Constraint*. Einzig diese beiden Constraints werden hier behandelt.

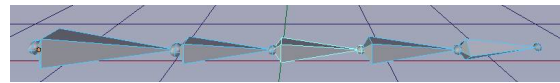
#### Action

Der Action Constraint ist ein mächtiges Instrument. Er erlaubt dir die Kontrolle über eine Aktion in Abhängigkeit von Transformationen eines anderen Objekts.

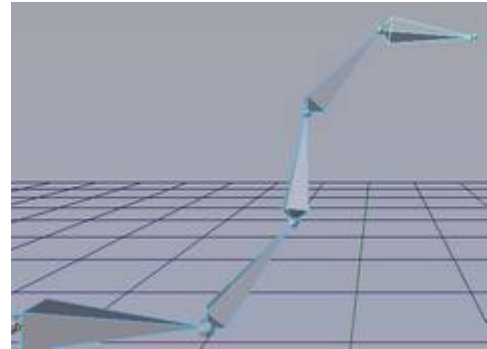
Die Aktion:

- Die Aktion wird vom Owner (gewöhnlich eine Knochenkette) ausgeführt.
- Die Aktion muss zunächst definiert werden: Eine Aktion besteht in einem Bewegungsablauf in der Zeit. Es gibt eine Ausgangsstellung und nach einer festgelegten Zeit eine Endstellung. Bei Blender wird dies mittels *Keyframes* in der *Timeline* festgelegt:

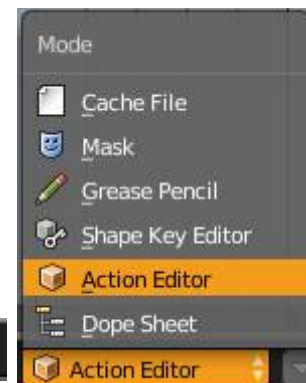
- Schaffe eine Knochenkette aus fünf Bones.



- Rufe in den Timeline Frame 1 auf.
- Selektiere die Knochenkette im Pose Mode und drücke [I] → LocRot.
- Rufe in den Timeline Frame 20 auf.
- Verändere die Position einzelner Knochen; selektiere die Knochenkette im Pose Mode und drücke [I] → LocRot.
- Wenn du jetzt die kleine Sequenz abspielst, richtet sich die Knochenkette aus der liegenden Position auf. **Das ist die Aktion.**
- Rufe jetzt zunächst den Dope Sheet Editor auf und dann den Action Editor (unterste Zeile).

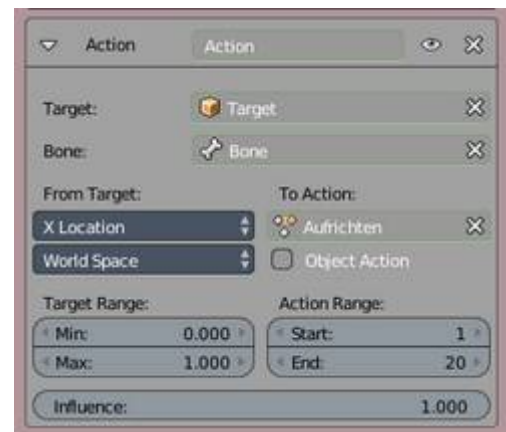


- Daneben findest du ein Namensfeld, in dem zunächst *OwnerAction*<sup>1</sup> steht. Klicke darauf und ändere diese Bezeichnung um in *Aufrichten*. **Jetzt hat die Aktion einen Namen.**



Einrichtung des Action Constraint:

- Wechsle wieder zum 3D Viewer und dort in den Pose-Mode.
- Die Verbindung mit dem Action Constraint muss für jeden Bone einzeln erfolgen:
  - Selektiere im Owner den ersten Knochen.
  - Rufe den Action Constraint auf.
  - Trage als *Target* den **Target** ein und als *Bone* den **Bone**.
  - Klicke in das Feld **To Action** und dort auf den



<sup>1</sup> Wenn du deine Knochen nicht (in Owner) umbenannt hast steht dort *ArmatureAction*.

Namen der Aktion: *Aufrichten*.

- Der **Action Range** kennzeichnet die Nummern der Frames, in der die Aktion stattfindet. Sie müssen unbedingt positiv und  $>0$  sein. Trage in der Spalte *Action Range* ein: **Start** = 1, **End** = 20.
- Der **Target Range** ist die Spanne, in der das Target wirksam wird. Bei ...*Location* sind dies Blendereinheiten, bei ...*Rotation* Werte zwischen  $-180^\circ$  und  $180^\circ$  und bei ...*Scale* Werte zwischen 0 und  $>0$ . Trage bei diesem Beispiel in der Spalte Target Range ein: Min = 0.00, Max = 1.00
- Wiederhole diese Schritte für alle Bones des Owners, die in den Vorgang einbezogen sind.
- **Test**
  - In der Spalte *From Target* steht standardmäßig im ersten Feld **X Location**. Das bedeutet, dass eine Bewegung des Target auf der X-Achse den Auslöser für unsere Aktion darstellt.
  - Selektiere das Target (im Pose- oder im Object-Mode) und bewege es auf der X-Achse. Wenn sich unsere Knochenkette aufrichtet, hast du alles richtiggemacht.
- **Andere Auslöser**
  - Im oberen Feld der Spalte *From Target* (Transform Channel) werden die verschiedenen Bedingungen aufgelistet, die zum Auslösen der Aktion führen können. Neben Verschiebungen des Target auf den drei Achsen können auch Rotationen oder Skalierungen verwendet werden. Da jedem Bone ein eigener *Action Constraint* zugewiesen werden kann, kann es zu unerwarteten Ergebnissen kommen, wenn einzelne Bones jeweils einen anderen Auslöser haben.
- Das Target muss kein Bone sein. Jedes Object (Mesh, Lamp Kamera und auch ein Empty) kann als Target fungieren. Was am günstigsten ist, hängt vom jeweiligen Zusammenhang ab.
- **Zu beachten:**
  - Wenn die Aktion Veränderungen der Location bewirkt, wird zu der gegenwärtigen Location des Owners die Werte *hinzuaddiert*, die sich aus der Aktion ergeben.
  - Wenn die Aktion Veränderungen der Skalierung bewirkt, wird die gegenwärtige Skalierung des Owners mit den Werten *multipliziert*, die sich aus der Aktion ergeben.
  - Wenn die Aktion Veränderungen der Rotation bewirkt, wird die existierende Rotation des Owners *überschrieben*.
  - Du kannst den Wert für *Start* höher setzen als den für *End* oder *Min* höher als *Max*. Dies wird den Ablauf der Aktion umkehren (rückwärts abspielen).
  - Leider kann dieser Constraint immer nur für einen einzelnen Knochen angewendet werden und nicht für eine Knochenkette oder eine Armatur als Ganzes. Alternativ kannst du die *pose library* anwenden, um einen ganzen Datenblock einer Pose herzustellen und zu editieren.

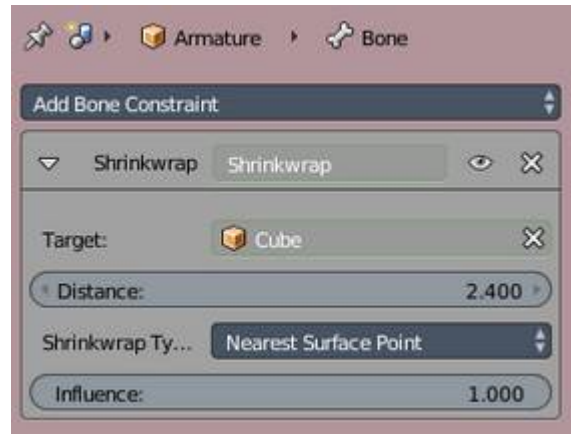


<https://vimeo.com/171554048>

## Shrinkwrap

Dieser Constraint bewirkt, dass ein Bone den Abstand zu einem Mesh nicht unterschreitet (vor dem Mesh gewissermaßen zurückschreckt).

- **Target** ist das Object, von dem der Bone Abstand hält. Dies muss ein Mesh sein.
- **Shrinkwrap Type** definiert die Methode des Abstandhaltes:
  - **Nearest Surface** wählt den nächsten Punkt auf der Oberfläche des Target.
  - **Nearest Vertex** wählt das nächste Vertex des Targets.
  - **Normal projection** projiziert Vertices in eine bestimmte Richtung (X, Y oder Z), bis sie das Target berühren.
  - **Distance** legt den Abstand fest, der von der Position des Target eingehalten werden muss.



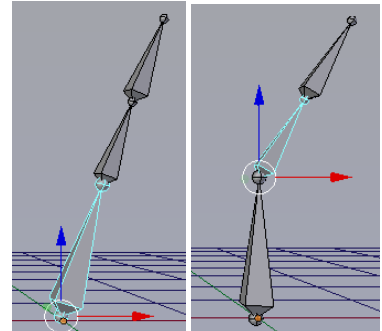
<https://vimeo.com/171554427>

### 7.4.4 Inverse Kinematics

Das Thema Inverse Kinematics ist so vielfältig und kann von so vielen Seiten betrachtet werden, dass es sinnvoll ist, hierzu ein eigenes Unterkapitel zu verwenden.

#### Der Grundgedanke

- Normalerweise geht die Bewegung in einer Knochenkette vom jeweiligen *Parent*-Bone aus und setzt sich dann über die Child-Bones usw. fort. Dieses Prinzip wird als *Forward Kinematic* bezeichnet.
  - Baue eine Knochenkette auf drei Bones auf, selektiere im Pose-Mode den untersten Knochen (Parent) und drehe diese mit [R] und Ziehen der Maus. Als Ergebnis bewegt sich die gesamte Kette, indem sie dem Parent folgt. (linkes Bild)
  - Wenn du den mittleren Knochen selektierst und diesen rotierst, bleibt der unterste Knochen ruhig und nur der obere wird mitbewegt. (rechtes Bild) Auch hier funktioniert die Bewegungssteuerung also vorwärts.
- Wenn du dir die Knochenkette eines Arms vorstellst, die sich von den Schultern ausgehend über Oberarm, Unterarm und Handgelenk bis in die Finger fortsetzt, wäre es allerdings sehr umständlich, in dieser Kette jeden einzelnen Knochen zu bewegen, um eine bestimmte Hand- bzw. Armstellung zu erzeugen. Einfacher ist es, wenn sich die Bewegung vom Ende, also von der Hand her steuern lässt. Dieses Prinzip wird als *Inverse Kinematics* (umgekehrte Bewegungssteuerung) bezeichnet. Hierbei orientieren sich die „vorgeschaleteten“ Knochen an der Position und Rotation eines Knochens, der weiter „oben“ in der Kette sitzt.

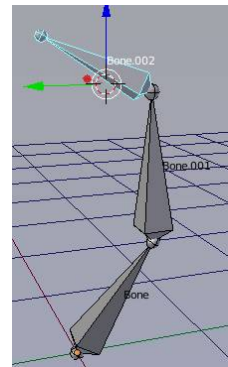


Um dies zu erreichen stehen, verschiedene Funktionen zur Verfügung:

## Automatic IK



- Wenn du im Pose Mode im Tool Shelf (links) die Registerkarte Options anklickst, erscheint unter *Pose Options* die Option **Auto IK**. Wenn du in das Kästchen davor ein Häkchen setzt, ist diese Funktion aktiviert.



- Wenn du jetzt den obersten Knochen selektierst und nach unten ziehst, bewegt sich die gesamte (eigentlich übergeordnete) Knochenkette mit.
- Die **Auto IK**-Funktion gilt für die gesamte Knochenkette. Wenn du Bone.001 selektierst, kannst du auch *Bone* mitbewegen, allerdings gilt für Bone.002 dann die Forward Kinematik.
- Insgesamt sind die Möglichkeiten von Auto IK recht eingeschränkt; in vielen Fällen reichen sie aber völlig aus.

## Inverse Kinematic (IK)

- **Beachte:** Bei der Nutzung von *Automatic IK* sind die steuernden Bones selbst Teil der Knochenkette. Das ist beim „normalen“ IK nicht zulässig. Dort gilt

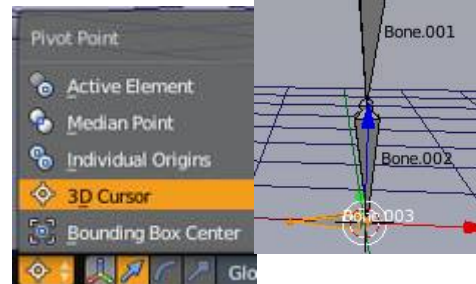
**Wenn ein Bone eine Hierarchie von Bones steuern soll, sollte dieser Bone nicht selbst Teil der Kette sein.**

Daher werden bei der Einrichtung von IK sog. Kontroll-Bones eingesetzt, mit denen die Steuerung erfolgen kann.

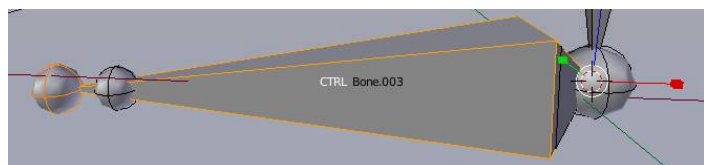
- Nehmen wir eine Bone-Kette mit vier Bones, wobei der erste Bone oben liegt und Bone.003 unten; entsprechend ist der Head jeweils oben und der Tail unten. Es könnte sich um die Armatur eines Beines oder eines Roboterarms handeln. Bone.003 ist eingeknickt, um einen Fuß anzudeuten.

- Installation des Steuerungsknochens (Kontroll-Bones):

- Wechsle in den Edit Mode und selektiere den Head von Bone.003.
- Platziere dort den Cursor: [shift] + [S] → **Cursor to Selected**.
- Lege fest, dass der Pivot-Punkt beim Cursor liegt.

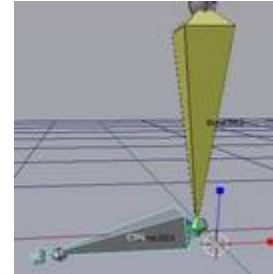


- Selektiere mit RMT Bone.003 als Ganzes und dupliziere ihn mit [shift] + [D], ohne das Duplikat an eine andere Stelle zu schieben. Verlängere jetzt den (duplizierten) Bone etwas nach vorn (mit [S] und Ziehen). Die kleinere Version ist jetzt also der Bone.003 und die größere Version der Steuerungsknochen. Gib diesem Bone den Namen *CTRL*.



- In den Bone-Eigenschaften kannst du erkennen, dass der Steuerungsknochen z.Zt. noch „Kind“ von *Bone.002* ist. Das soll er aber nicht. Drücke also [alt] + [P] und klicke auf **Clear Parent**. Damit ist diese Verbindung gelöscht.
- Jetzt kann der **Inverse Kinematic Constraint** im **Pose Mode** festgelegt werden:

- Wähle den Steuerelementknochen mit RMT aus (das ist das Target).
  - Wähle den Bone.002 mit [shift] + RMT aus.
  - Drücke [shift] + [strg] + [C] und wähle dann in der Spalte **Tracking Inverse Kinematics** aus.
  - Jetzt wird Bone.002 gelb eingefärbt, denn für diesen Bone gilt der Constraint.
- Wenn du jetzt CTRL selektierst und bewegst, wirst du feststellen, wie sich diese Bewegung über die gesamte Knochenkette fortsetzt.



## Bewegungsmöglichkeiten einschränken

- Du kannst dir u.U. viel Arbeit ersparen, wenn du für jeden Knochen festlegst, in welchen Achsen und Winkeln jeder einzelne Knochen bewegt werden kann. Das geht allerdings nur für die Knochen, die in eine Inverse Kinematics-Kette eingebunden sind.
- Selektiere also *Bone.002* und öffne dann *Bone* (Knochen-Icon) → *Inverse Kinematics*.
- Links sind die drei Bewegungsachsen X, Y und Z aufgeführt. Für jede Achse kannst du festlegen, dass die möglichen Bewegungen limitiert sind, indem du ein Häkchen vor **Limit** setzt. Damit kannst du dann auch in den zugehörigen Feldern eintragen, um wieviel Grad die Bewegung zulässig ist.
- Trage beispielsweise für die X- und die Z-Achse die Werte der Limits ein. Um deinen Knochen erscheinen dann die sich daraus ergebenden Begrenzungsschalen. Der Knochen kann sich z.B. um die X-Achse sehr weit nach vorn und hinten bewegen. Um die Z-Achse hingegen nur sehr wenig. So wirken sich die vorgenommenen Einstellungen auf die Bewegungsmöglichkeiten aus:
- Zu Erläuterung der Parameter:
  - **Lock** X, Y Z (Schloss-Symbol): Sperrung der Drehbewegung des Knochens um eine oder mehrere der drei Achsen.
  - **Stiffness**: Starrheit der Bewegung in der IK-Kette.
  - **Limit** – Häkchen und Winkelangabe: Begrenzung der Drehbewegung um die jeweilige Achse.

