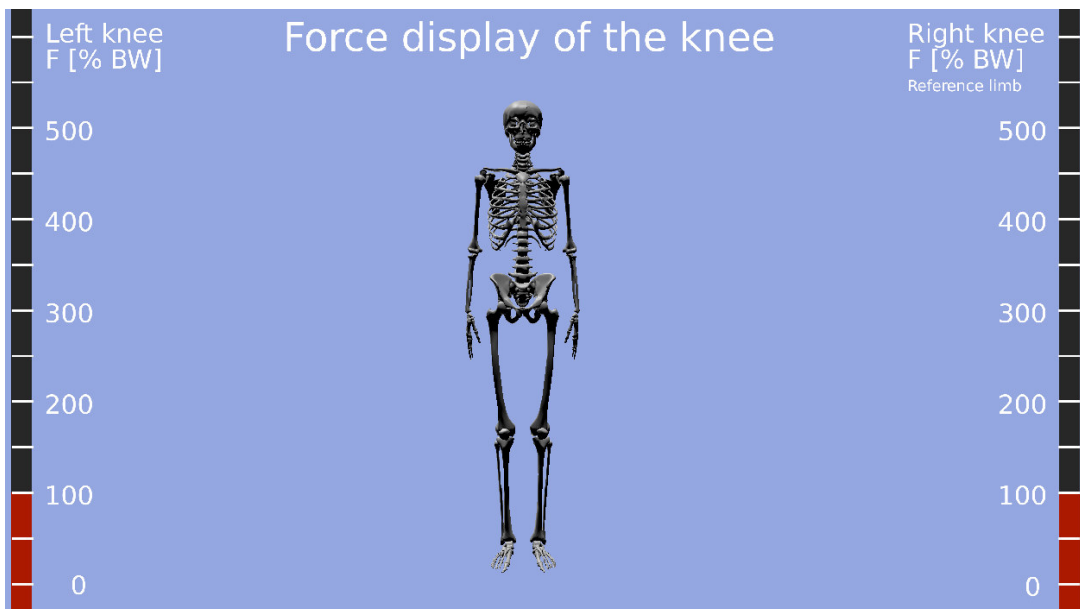
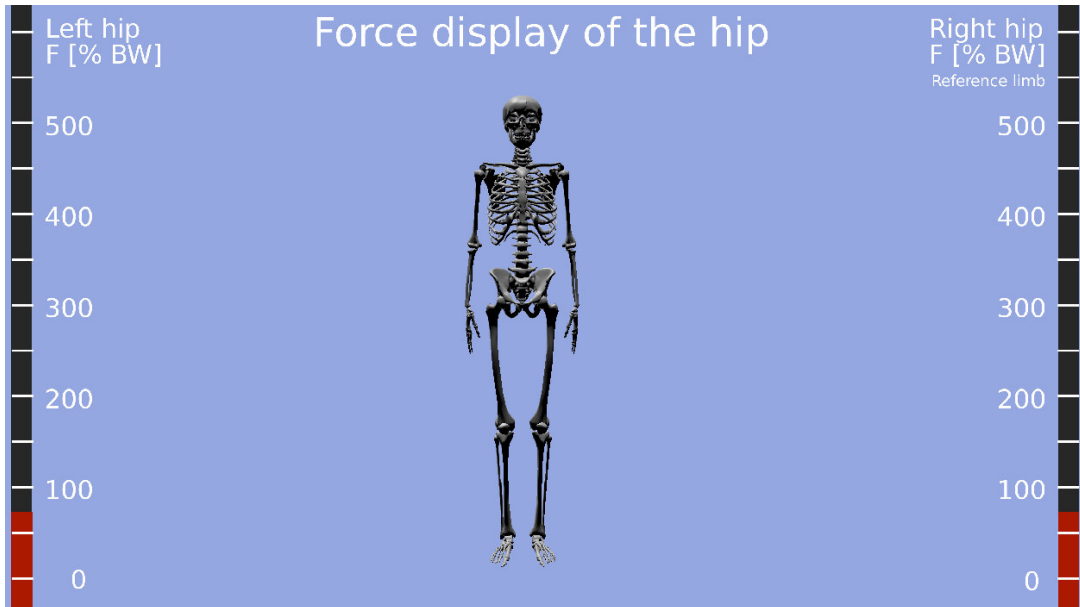


Konstruktionsplan

für das Programm „Motion and force display“



Erstellt von Weronika A. Krzywik-Groß und Gerwin Reimers.

Vorwort

Die vorliegende Arbeit umfasst sämtliche Konstruktionsvorgänge für das Programm „Motion and force display“. Die Arbeitsanweisungen werden in einer Kurzform gehalten.

Die Darstellung der Kräfte am Hüft- und Kniegelenk erfolgt mit dem Programm Blender 2.73. Im Jahr 2002 wurde das 3D-Modellierungs- und Animationsprogramm als eine Open-Source-Software (GPL) veröffentlicht (Beck 2015, S. 24).

Das in dieser Arbeit verwendete Skelett wurde bei SC Computer Graphics Studio Srl erworben. Das Modell ist nach physiologischen Gesichtspunkten erstellt worden, besitzt insgesamt 102788 Polygone und 52911 Vertices (SC Computer Graphics Studio Srl 2015).

Die Software Blender wurde mit den Programmiersprachen Python und C++ erstellt. Da sich die Programmiersprachen als auch die Software im englischen Sprachraum bewegen, wird auf eine Umstellung auf die deutsche Sprache verzichtet. Das hat den Vorteil, dass Missverständnisse bei Übersetzungsvorgängen minimiert werden.

Inhaltsverzeichnis

1 Die Arbeitsoberfläche	4
2 Positionierung des Skeletts im 3D-Raum	4
3 Erstellen von Knochengruppen (Join-Verbindungen)	6
4 Erstellung einer Armature (Rigging)	9
5 Die Achsenausrichtung	17
6 Das Skinning	18
7 Die Control-Bones	19
8 Constraints	21
8.1 Die Kopf – Nacken – Bewegung	22
8.2 Die Armbewegung	23
8.3 Die Becken- und Oberkörperbewegung	24
8.4 Die Beinbewegung	24
9 Die Ebenen	26
10 Die Bone Groups	27
11 Das Custom Shape	28
12 Animation des Bewegungsablaufs	34
13 Hintergrund	45
14 Untergrund Game Engine	46
15 Kamera und Licht	47
15.1 Kamera	47
15.2 Licht	50
16 Erstellung der Kraftanzeige	50
17 Break	54
18 Animation der Kraftanzeige vom Hüftgelenk	55
19 Start-Datei	60
20 Game Engine	66

21 Knee-Datei	79
22 Programmerstellung	81
Abbildungsverzeichnis	83
Tabellenverzeichnis	88
Literaturverzeichnis	89
Internetverzeichnis für verwendete Blenderfunktionen	90

1 Die Arbeitsoberfläche

Die Blender-Arbeitsoberfläche (Abb. 1) wird in folgende Kategorien eingeteilt:

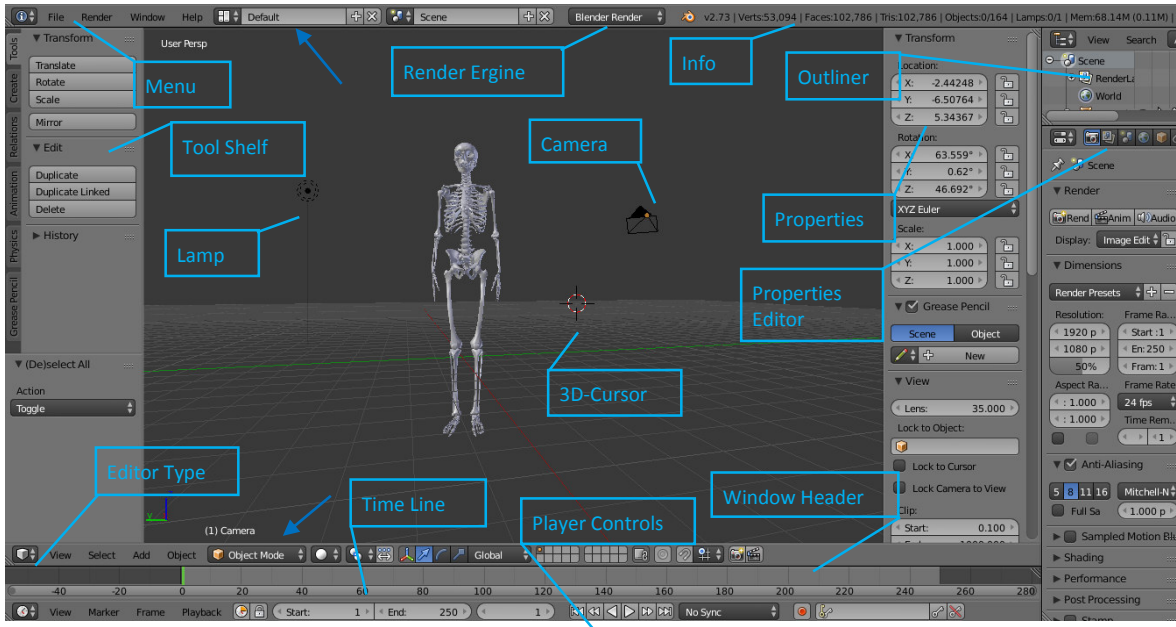


Abb. 1: Arbeitsoberfläche von Blender
(Wartmann 2011, S. 33)

Die ersten Umsetzungen erfolgen im Default-Layout sowie im Object Mode (Abb. 1).

2 Positionierung des Skeletts im 3D-Raum

Das Ziel ist das Skelett im Zentrum des 3D-Raumes zu positionieren. Zunächst wird mit **[Pad 5]** von der perspektivischen in die orthographische und mit **[Pad 1]** in die frontale Ansicht gewechselt. Mit **[A]** wird das ganze Modell selektiert, mit **[R] + [Z]** in die frontale Position rotiert und mit **[G] + [X]** in die Mitte bewegt (Abb. 2).

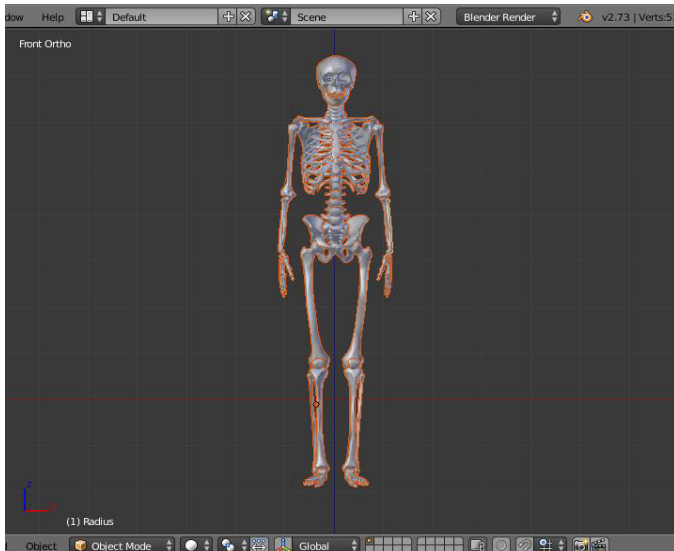


Abb. 2: frontal-zentrierte Positionierung des Skeletts

Das Skelett wird über **Pad3** in die Seitenansicht gebracht und mit **G** + **Z** und **G** + **Y** auf die y-z-Achse verschoben. Mit **Pad1** wird in die frontale Ansicht zurück gewechselt.

Ein gemeinsamer Orientierungspunkt für alle Knochen wird erstellt. Der 3D-Cursor wird mit **Alt** + **C** im Ursprungspunkt positioniert. Im Tool Shelf im Tab Tools unter Set Origin Origin to 3D-Cursor auswählen und mit **A** die Markierung aufheben. Mittels dieser Einstellung haben alle Knochen des Skeletts einen gemeinsamen Orientierungspunkt (Abb. 3), den sogenannten Pivot-Punkt. Alle Skalierungen, Rotationen und Spiegelungen bewegen sich um diesen Pivot-Punkt im 3D-Raum (Beck 2015, S. 93).

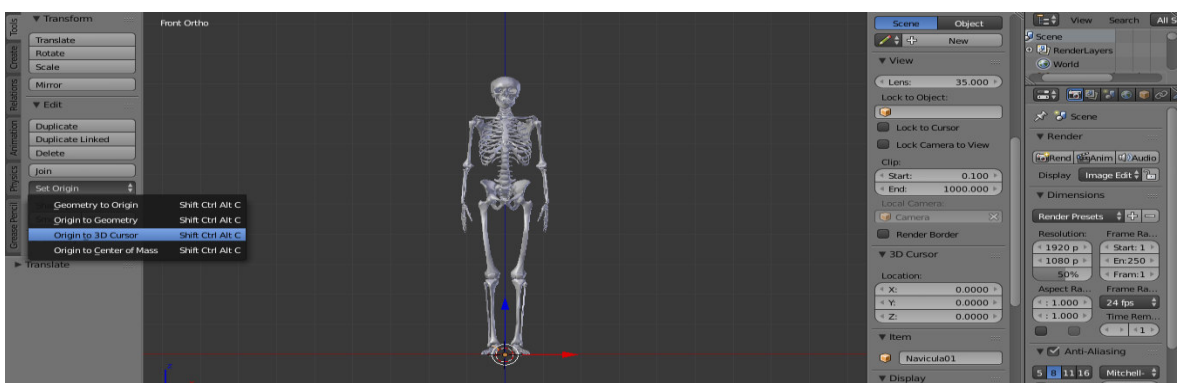


Abb. 3: gemeinsamer Orientierungspunkt der Skelettknochen

Blender arbeitet mit einer eigenen Längeneinheit BU (Blender Unit = Blendereinheit) und geht von einem Umrechnungsverhältnis $1\text{m} = 1\text{ BU}$ aus (Beck 2015, S. 97).

Die Blendereinheit wird in Zentimeter umgewandelt. Im Properties Editor wird im Icon Scene unter Units Metric eingestellt. Mit \boxed{S} wird die Größe auf 1,62m skaliert (Abb. 4).

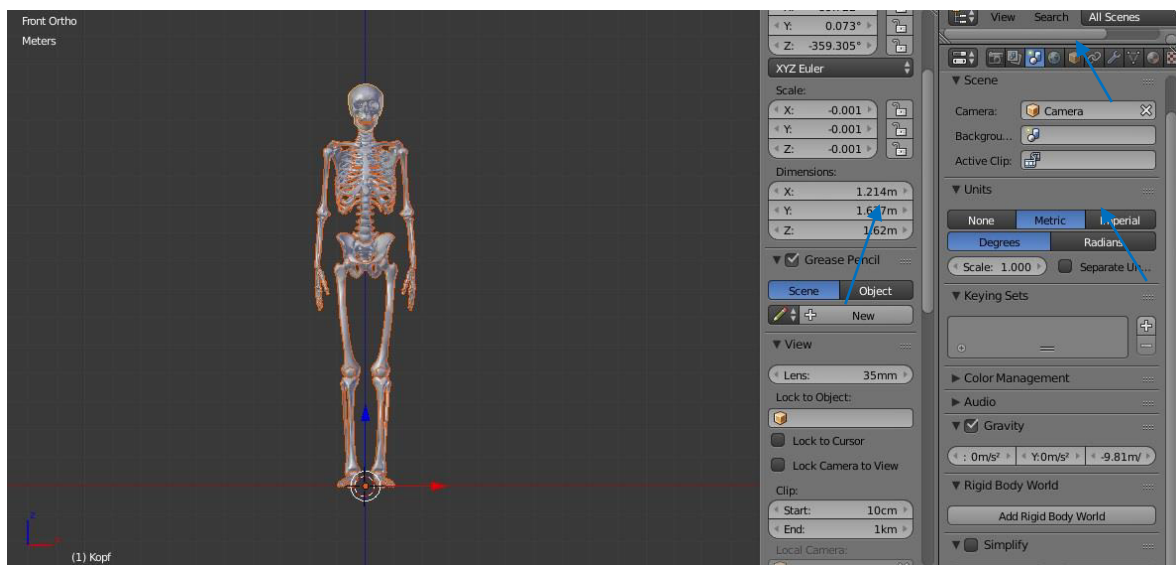


Abb. 4: Skalierung des Skeletts

3 Erstellen von Knochengruppen (Join-Verbindungen)

Die Knochen des Skeletts (Mesh¹) stehen in keiner Abhängigkeit zu einander und werden als Einzelobjekte betrachtet. Um das Rig (Kap. 4), und später die Animation (Kap. 12) zu vereinfachen, werden einzelne Knochen vom Mesh zu gemeinsamen Gruppen zusammengefügt. Im Object Mode werden zunächst die Knochen, die zusammen gehören sollen, mit \boxed{A} + $\boxed{\text{Shift}}$ selektiert. Anschließend werden diese mit $\boxed{\text{Strg}}$ + \boxed{J} vereint.


¹ Unter Mesh werden alle Objekte, die über Vertices, Edges sowie Faces verfügen, verstanden (Beck 2015, S. 121).

Die Knochen des Skeletts werden wie in Tabelle 1 in folgenden Gruppen vereint (Schünke M. et al. 2007, S. 38):

<i>Skelettknochen</i>	<i>Join-Elemente</i>
Zehenknochen	Ossa digitorum pedis
Mittelfußknochen	Ossa metatarsi
Fußwurzelknochen	Ossa tarsi
Schienbein und Wadenbein	Tibia
Kniescheibe	Patella
Oberschenkelknochen	Femur
Beckenknochen	Pelvis
Kreuzbein, Steißbein	Sacrum
Lendenwirbel	Vertebrae lumbales
Brustkorb	Skeleton thoracis
Halswirbel	Vertebrae cervicales
Schädel + Zähne	Cranium
Oberarmknochen	Humerus
Speiche und Elle	Radius
Handwurzelknochen	Ossa carpi
Mittelhandknochen	Ossa metacarpalia

Fingerknochen	Ossa digitorum manus
Daumen (erster Fingerknochen)	Phalanx proximalis
Daumen (zweiter Fingerknochen)	Phalanx distalis

Tabelle 1: Join-Elemente der Skelettknochen

In Blender sind die einzelnen Knochen sowie die Join-Verbindungen des Skeletts im Outliner-Menü anatomisch bezeichnet. Nach Bedarf können die Join-Verbindungen umbenannt (Tab. 1) werden. Mit  die nun definierte Gruppe auswählen. Dabei wird im Outliner-Menü das Join-Element hervorgehoben. Mit einem Doppelklick diesen Namen anwählen und betiteln. Es empfiehlt, aufgrund der besseren Zuordnung, sich die Körperseiten mit "links" oder "rechts" zu deklarieren (Abb. 5).

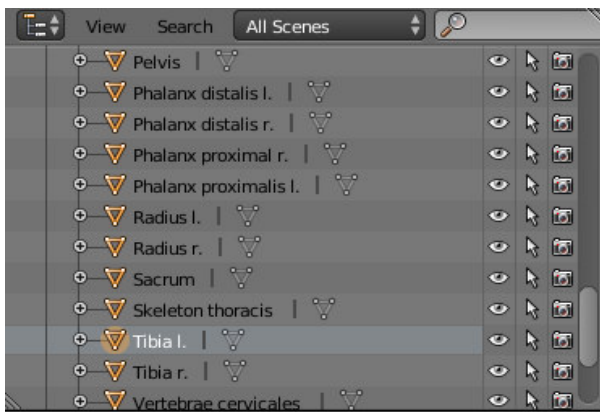


Abb. 5: Benennung der Join-Elemente im Outliner

4 Erstellung einer Armature (Rigging)

Für die spätere Animation (Kap. 12) ist es notwendig das Skelett-Modell mit einem Rig (Knochenskelett) auszustatten. Durch das Riggen kann dieses bewegt werden. Beim Riggen werden dem Skelett über das sogenannte Armature-Werkzeug einzelne Bones (Knochen) hinzugefügt. Mehrere miteinander verbundene oder voneinander getrennte Bones (Abb. 6) werden Armature genannt (Beck 2015, S. 511).

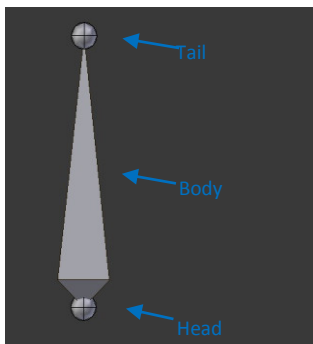


Abb. 6: Bone

Im Object Mode mit **B** + das Skelett markieren und mit + **S** im Snap-Fenster Cursor to Selected wählen (Abb. 7). Der Cursor erscheint am Pivot-Punkt.

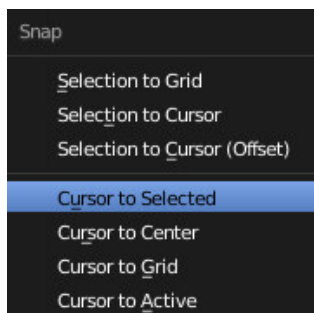


Abb. 7: Snap - Fenster

Mit dem Armature-Werkzeug mittels + **A** Single Bone auswählen. Am 3D-Cursor entsteht ein Bone. In einem Mesh sind Bones nicht sichtbar. Um Bones in die richtige Position zu bringen und später bearbeiten zu können wird daher im Properties Editor unter dem Icon Object Data im Panel Display X-Ray aktiviert (Abb. 8).

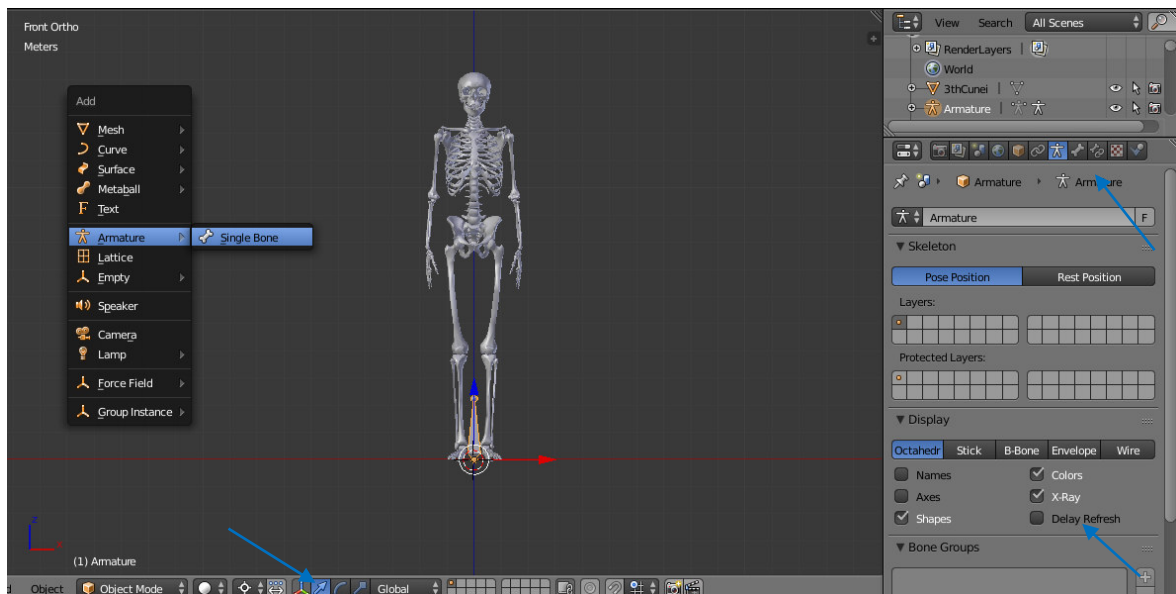


Abb. 8. Erstellung eines Bones

Bei der Erstellung aller Bones ist es sinnvoll die Positionierung der Bones mit $\text{G} / \text{R} + \text{X} / \text{Y} / \text{Z}$ oder wahlweise mit dem Transformation Manipulator Translate (Abb. 8) mittels $\text{G} + x- / y- / z$ -Koordinaten durchzuführen. Welche Option gewählt wird, entspricht dem individuellen Arbeitsstil.

Bezüglich der erfolgten Ausrichtung bleiben die Bones in gleicher Achsenorientierung. Dieses ist für die Funktionalität der Constraints (Kap. 8) sowie der Steuer-Bones (Kap. 7) und die nachfolgende Animation (Kapitel 12) notwendig.

Vom Object Mode wird über E in den Edit Mode gewechselt², anschließend den erzeugten Bone selektieren und diesen mit $\text{G} + \text{Z}$ (oder wahlweise mit $\text{G} + z$ -Koordinate) in Rückenhöhe verschieben. Den Bone auf die Gesamtlänge der Wirbelsäule entsprechend skalieren. Hierzu wird zunächst mit H Head des Bones markiert und mit $\text{S} + \text{Z}$ zum Steiß, anschließend wird Tail selektiert (H) und zum ersten Halswirbel gezogen ($\text{S} + \text{Z}$).

Den ganzen Bone auswählen (A) und im Tab Tools Subdivide ausführen. Der erstellte Bone wird in insgesamt vier Bones unterteilt, für jedes Join-Element einen Bone (Sacrum, Vertebrae lumbales, Skeleton thoracis, Vertebrae cervicales). Die

² Alternativ kann im Window Header der Wechsel zwischen Object Mode und Edit Mode manuell erfolgen.

einzelnen Bones werden am Tail/Head markiert (☞) und mit ☞ + x-Koordinate in die Stellung der Join-Verbindungen gebracht. Zur genauen Positionierung der Bones an den Join-Elementen kann zwischen dem Edit Mode und dem Object Mode mit ☞ gewechselt werden. Deren Position aus jeder Perspektive überprüfen, so dass die Bones an dem jeweiligen Mesh des Skeletts platziert sind.

Der Bone am Sacrum-Join-Element wird mit Head zentral zwischen die beiden Femur-Knochen platziert. Zur genauen Ausrichtung kann ein Hilfs-Bone³, entstanden im proximalen zum zweiten proximalen Femur und mit Subdivide in zwei Bones unterteilt, verwendet werden. Zur exakten Positionierung der Hilfsbones kann mithilfe von ☞ im Object Mode das Becken ausgeblendet⁴ und die Bone-Form im Properties Editor unter Object Data in Sticks umgewandelt werden (Abb. 9). Die zentrale Positionierung des Bones wird in der Animation für die Positionierung des Beckens von Bedeutung.

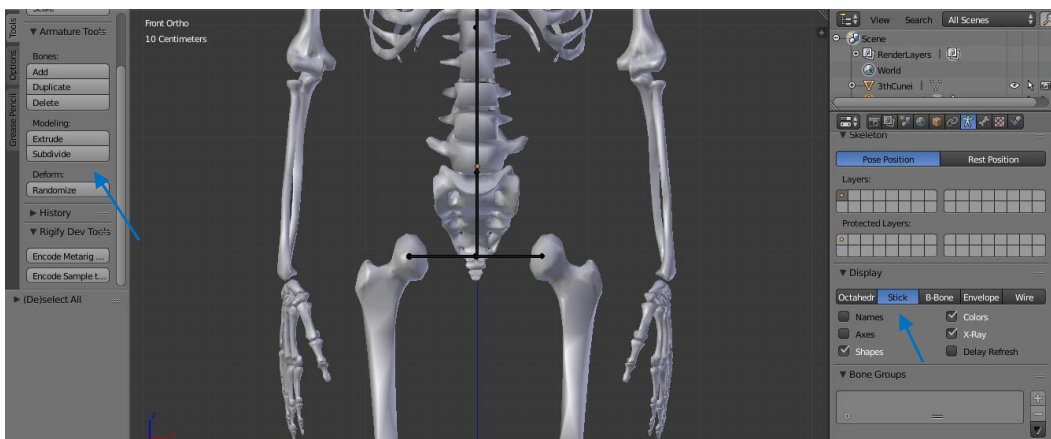


Abb. 9: Hilfs-Bones zur Erstellung des Sacrum-Bones

Die Hilfs-Bones werden anschließend wieder mit ☞ entfernt, die Form der Wirbelsäulen-Bones in die Oktaeder-Form gebracht.

Aus dem Tail vom Bone am Vertebrae cervicales (☞) wird ein Bone für den Kopf mit ☞ + ☞ extrudiert und am Schädel positioniert (Abb. 10).

³ 3D-Cursor im proximalen Femur positionieren, mit ☞ + ☞ einen Bone erschaffen.

⁴ Das Wiedereinblenden vom Becken erfolgt mit ☞ + ☞.

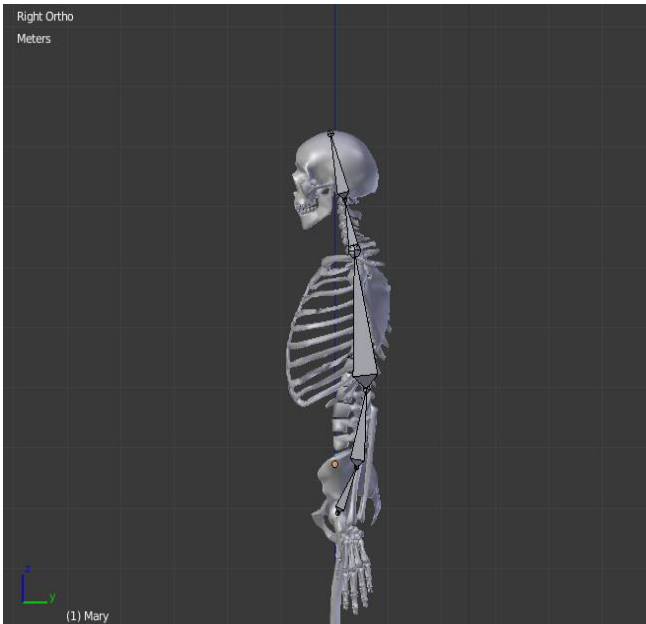


Abb. 10: Armature an der Wirbelsäule und Kopf

Die Armature für die linke und die rechte Seite der Arme und Beine werden jeweils einzeln erstellt.⁵ Für die Arme wird aus dem Tail vom Bone am Skeleton thoracis mit **E** + **X** ein Bone in die x-Richtung extrudiert. Dabei entsteht automatisch eine Parent-Child-Verbindung. Mit **Alt** + **P** und Clear Parent wird ebendiese gelöscht (Abb. 11).

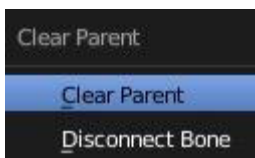


Abb. 11: Clear Parent – Fenster

⁵ Blender verfügt über einen "Mirror-Modifier". Bei diesem Vorgang wird nur eine Seite des Objektes erstellt. Diese Seite kann dupliziert und dann gespiegelt werden. Dabei entsteht eine zweite Seite des Objektes ohne eine weitere Konstruktion. Der "Mirror-Modifier" erzeugt demzufolge spiegelsymmetrische Meshes/Objekte (Beck 2015, S. 216). Anatomisch gesehen ist der menschliche Körper nicht spiegelsymmetrisch. Das Skelett wurde demgemäß konstruiert. Daher kann "Mirror-Modifier" hier nicht verwendet werden.

Der erzeugte Bone wird mit dem Head an dem proximalen und mit dem Tail an dem distalen Humerus-Knochen platziert. Dazu wird der Bone-Tail mit **R** + **9** **0** um 90° vertikal rotiert, mit dem Transformation Manipulator Translate in x-, y- und z-Richtung gesteuert und mit **S** skaliert. Zur Erzeugung einer neuen Parent-Child-Verbindung zwischen dem neu positionierten Bone und dem Bone am Vertebrae cervicales muss zunächst das Kind-Objekt, hier der Bone am Humerus, und anschließend das Elternobjekt, hier der Bone am Vertebrae cervicales, selektiert (**⇧** + **⇩**) werden. Mit **Strg** + **P** wird im Fenster Make Parent und Keep Offset die Parent-Child-Verbindung erstellt (Abb.12). Dabei wird Head des Kindes mit dem Tail des Eltern-teils verbunden, wobei das Kind an seiner ursprünglichen Position bleibt. Als Zeichen für die erfolgte Parent-Child-Verbindung entsteht zwischen den beiden Bones eine gestrichelte Linie.

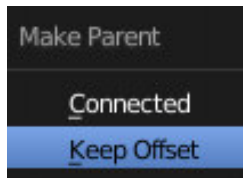


Abb. 12: Make Parent – Fenster

Mit **E** + **X**⁶ werden die nötigen Bones für den Arm extrudiert: Radius, Ossa carpi, Ossa metacarpalia, Ossa digitorum manus. Für den Daumen (Phalanx proximalis, Phalanx distalis) werden zwei aufeinanderfolgende Bones aus dem Bone am Ossa carpi extrudiert. Die Stellung der einzelnen Bones aus allen Ansichten in die Position am Mesh mit **R**, **S** und Transformation Manipulator Translate in x-, y- und z-Richtung ausrichten.

In Äquivalenz dazu, wird der eben erfolgte Vorgang am anderen Arm wiederholt. Die Software erstellt die Bones in numerischer Reihenfolge, die linke von der rechten Seite unterscheidend mit .001. Die Bones werden im Edit Mode einzeln markiert (**⇧**) und im Properties unter Item umbenannt (Tab. 2). Das Armature kann hier einen

⁶ Um die Achsausrichtung beizubehalten, müssen die weiteren zu extrudierenden Bones gleicher Richtung des zuerst extrudierten Bones erstellt werden.

eigenen Namen erhalten (Abb. 13). Vorausgesetzt, dass Properties im Hintergrund ist, wird dieses mit **N** sichtbar.

Join-Elemente	Bone-Benennung
Sacrum	Sacrum
Vertebrae lumbales	Spine
Skeleton thoracis	Ribcage
Vertebrae cervicales	Neck
Cranium	Skull
Humerus	Humerus
Radius	Radius
Ossa carpi	Ossa carpi
Ossa metacarpalia	MC
Ossa digitorum manus	Phalanges
Phalanx proximalis	Phalanx proximalis
Phalanx distalis	Phalanx distalis

Tabelle 2: Join-Elemente des Oberkörpers

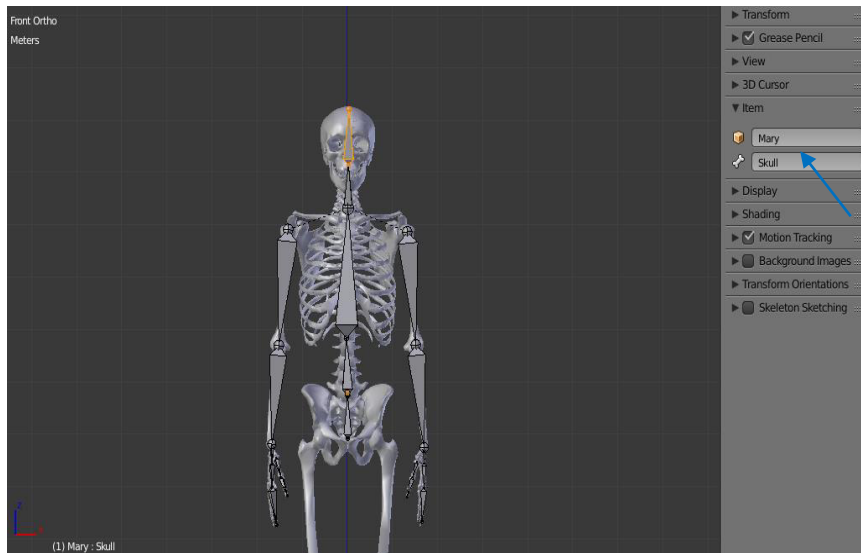


Abb. 13: Armature am Oberkörper, Benennung der Bones

Am Bein wird mit dem Bone an der Patella begonnen. Hier wird im Edit Mode ein Bone im Zentrum der Patella erzeugt ($\square + A$). Der Bone wird zum distalen Femur mit Hilfe von Transformation Manipulator Translate in x-, y- und z-Richtung positioniert, so dass dieser waagrecht zum Femur liegt und der Head nach dorsal zeigt. Der Patella-Bone wird etwas größer konstruiert (Abb. 14). Dieser Bone wird mit $\square + D + X$ dupliziert und zum Zentrum der zweiten Patella verschoben. Über die Tasten $D + X$ wird der Bone automatisch auf der x-Achse verschoben.

Für das Bein wird ein Bone am proximalen Femurende erschaffen ($\square + A$). Der erzeugte Bone wird im Zentrum vom Femurkopf platziert. Zur genauen Ausrichtung des Bones wird die Wireframe-Ansicht im Viewport aktiviert (Abb. 15). Seine Positionierung am Femur gleicht der mechanischen Beinachse. Eine Keep Offset Verbindung wird zwischen dem Bone am Femur und dem Sacrum-Bone erstellt.

Für die Bewegung des Kniegelenks werden aus dem Bone-Tail am Femur zwei aufeinanderfolgende Extrusionen in x-Richtung ($E + X$) durchgeführt. Der erste Bone wird nach dorsal, und von diesem der zweite Bone zurück zum Bone-Tail am Femur positioniert. Das ist für die spätere Simulation der Roll-Gleitbewegung des Kniegelenks notwendig.

Aus dem zu Letzt erstellten Bone folgt über die Tastenabfolge ($E + X$) ein weiterer Bone für die Tibia. Der Bone an der Tibia wird ebenfalls am Mesh ausgerichtet (Abb.

14). Mit **Strg** + **P** und Keep Offset wird eine Parent-Child-Verbindung zwischen dem Bone an der Patella und dem Bone an der Tibia erstellt.

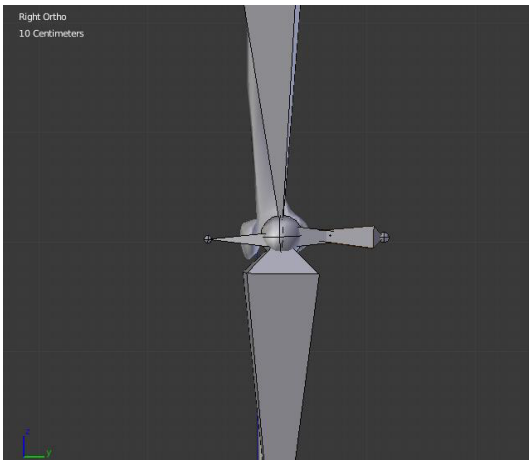


Abb. 14: Armature am Kniegelenk

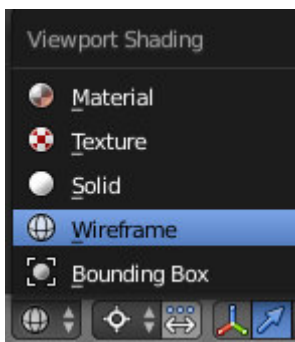


Abb. 15: Viewport Shading – Fenster

Bei der Konstruktion der Bones am Kniegelenk ist zu beachten, dass trotz der nicht bestehenden Connected-Parent-Child-Verbindung zwischen dem Bone am Femur und dem Bone an der Tibia, der Femur-Tail und der Tibia-Head in demselben Punkt ausgerichtet sind.

Für den Fuß werden drei Bones für die jeweiligen Join-Elemente (Ossa digitorum pedis, Ossa metatarsi, Ossa tarsi) erstellt, beginnend mit einer Extrusion am Tail an dem Bone an der Tibia (**E** + **X**). Mithilfe vom Transformation Manipulator Translate in x-, y- und z-Richtung werden die erstellten Bones am Mesh ausgerichtet.

In Äquivalenz dazu, wird der eben erfolgte Vorgang bei am anderen Bein wiederholt.

Im Properties unter Item werden die Bones für die unteren Extremitäten wie in Tabelle 3 umbenannt:

Join-Elemente	Bone-Benennung
Femur	Femur
Patella	Patella
Tibia	Tibia
Ossa tarsi	Heel
Ossa metatarsi	Metatarsus
Ossa digitorum pedis	Toe






Tabelle 3: Join-Elemente der unteren Extremitäten

5 Die Achsenausrichtung

Anhand der bisher erfolgten Konstruktion sollte die Achsenausrichtung aller Bones in gleicher Richtung sollte übereinstimmen. Diese kann ergänzend im Properties Editor unter Object Data im Display Axes überprüft werden. Aufgrund dieser Einstellung werden die Achsen aller Bones sichtbar. Alle Bones in einer Richtungslinie, beispielsweise alle Bones am Bein, sollten die gleiche Richtung aufweisen. Falls dem nicht so ist, können die Achsen der Bones im Edit Mode korrigiert werden. Dazu wird der Bone ausgewählt und unter **Strg** + **N** die globale Achse verwendet.

6 Das Skinning

Für die Bewegung des Skeletts über das Armature muss das Armature mit dem Mesh des Modells verbunden werden. Durch das sogenannte Skinning wird ein Mesh einem Bone zugewiesen. Damit wird bestimmt, welche Bones welchen Teil des Skeletts steuern sollen (Beck 2015, S. 35).

Im Pose Mode⁷ wird das Mesh markiert ()⁷). Dabei wechselt Blender automatisch in den Object Mode. Dort den zugehörigen Bone markieren ( + ). Das hat zur Folge, dass Blender automatisch zurück in den Pose Mode wechselt. Anschließend wird über die Tasten  +  im Set Parent To-Fenster Bone gewählt (Abb. 16). Mit der Auswahlmöglichkeit Bone wird das Mesh an den ausgewählten Bone gebunden, ohne dass die Armature beim Bewegen des Meshes diesen verformt.

Dieser Vorgang wird für jedes Mesh, wie es die Join-Elemente vorgeben, einzeln durchgeführt. Eine Ausnahme bilden das Sacrum- und das Pelvis-Mesh. Beide werden über den Sacrum-Bone gesteuert.

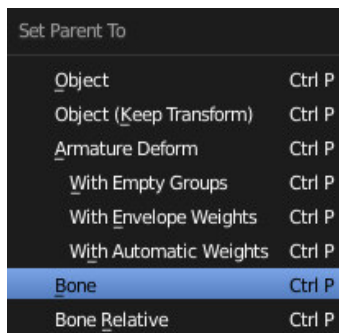


Abb. 16: Set Parent To – Fenster

⁷ Der Wechsel in den POSE MODE erfolgt manuell über den Window Header.

7 Die Control-Bones

Mit den Steuer-Bones (CTRL-Bones) wird das Skelett zukünftig gesteuert. Diese Bones dürfen nicht Bestandteil einer Kette sein. Es werden insgesamt sechs CTRL-Bones erstellt: einer für die Steuerung der unteren Extremitäten, zwei für die Knie und Oberschenkel, zwei weitere für die inverse Beinführung der Tibie und ein Haupt-CTRL-Bone, der das ganze Armature lenkt.

Im Edit Mode wird am Head des Sacrum-Bones ein CTRL-Bone konstruiert. Dazu wird der Head-Sacrum selektiert, mittels $\square + \text{S}$ wird im Fenster Snap Cursor to Select sowie im Window Header unter 3D Cursor Bounding box center ausgewählt. Anschließend wird mit der Tastenkombination $\square + \text{A}$ ein Bone erstellt. Dieser Bone bleibt mit dem Tail im 90°-Winkel (ganzen Bone auswählen, $\text{R} + \square \text{9} \text{0}$, mit der y-Koordinate positionieren) direkt am Sacrum-Bone positioniert. Eine Parent-Child-Verbindung zum Sacrum-Bone ist nicht vorhanden. Dieser CTRL-Bone ist für die Steuerung der unteren Extremitäten zuständig.

Für die Kniestuerung wird am Femur-Tail ein Bone erstellt. Dieser Bone wird selektiert und mit $\text{R} + \square \text{9} \text{0}$ um 90° gekippt. Anschließend wird der Bone auf der Knie-Achse mit \square und der y-Koordinate nach vorn verschoben. Dabei ist der Tail nach vorne gerichtet. Der neu erstellte Bone mit den Tasten $\square + \text{D} + \text{X}$ duplizieren und parallel auf der x-Achse zum zweiten Knie verschieben.

Für die inverse Beinführung wird ein Bone, der als CTRL-Bone für einen Inverse-Kinematics-Constraint (IK-Constraint, Kap. 8) dienen wird, soll direkt am Heel-Bone erstellt werden. An den Heel-Bones werden jeweils ein Bone dupliziert ($\square + \text{D}$) und die Parent-Child-Verbindung mit $\text{Alt} + \text{P}$ und Clear Parent gelöscht. Der erstellte Bone bleibt direkt am Heel-Bone und wird im Properties unter Item mit dem Kürzel IK umbenannt. Zur Unterscheidung für die späteren Bearbeitungen der übereinander liegenden Bones wird der Bone mehrmals angewählt (\square). Unter Item wird der Bone mit dem dazugehörigen Namen angezeigt, welcher gerade selektiert wurde.

Der Haupt-CTRL-Bone, mit dem das ganze Armature später gesteuert wird, soll auf dem Boden im Ursprungspunkt erstellt werden. Dazu wird über die Tastenkombination $\text{Alt} + \text{C}$ der Cursor am Ursprungspunkt positioniert und mit $\square + \text{A}$ der Haupt-

CTRL-Bone erzeugt. Dieser wird selektiert und um 90° (**R** + **9 0**) gekippt. Dabei zeigt der Tail nach vorne in die y-Richtung.

Die erstellten Bones werden im Properties unter Item mit dem Kürzel CTRL vorweg umbenannt.

Damit das Skelett durch die CTRL-Bones nicht verformt wird, muss im Edit Mode im Properties Editor unter Bone die Auswahl Deform ausgeblendet werden (Abb. 17). Diese Einstellung wird an jedem CTRL- und IK-Bone einzeln vorgenommen.

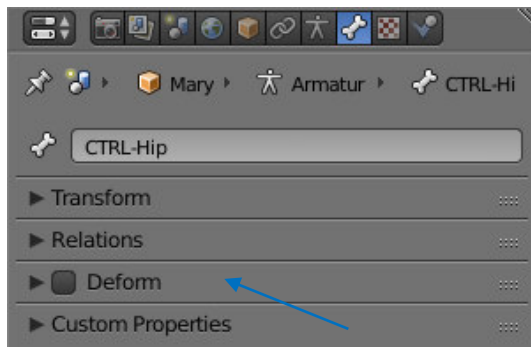


Abb. 17: Deform – Deaktivierung

Die CTRL- und IK-Bones werden über eine Keep Offset-Verbindung mit dem Haupt-CTRL-Bone verbunden. Die Bones werden nacheinander, zuletzt der Haupt-CTRL-Bone, selektiert (**Alt** + **U**). Mittels der Tastenkombination **Strg** + **P** wird die Verbindung erstellt (Abb. 18).

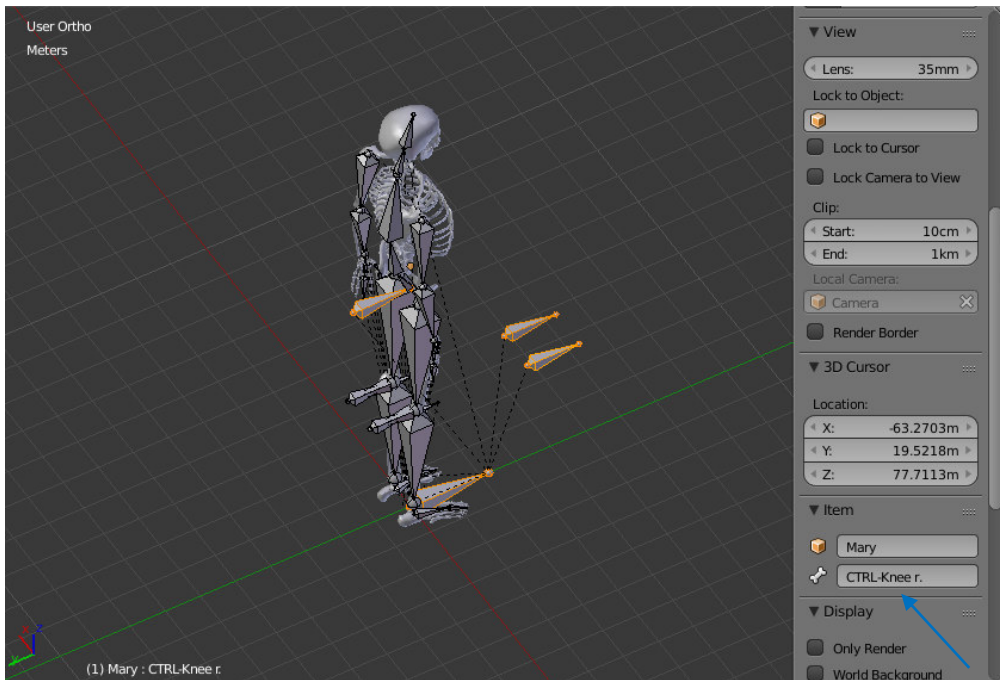


Abb. 18: CTRL- und IK-Bones in Verbindung mit dem Haupt-CTRL-Bone

Für die Steuerung der Arme werden sogenannte Constraints verwendet (Kap. 8).

8 Constraints

Mittels den sogenannten Constraints werden Objekte in Abhängigkeit von einem Zielobjekt (Target) gesetzt. Die Objekte können beispielsweise in ihrer Bewegung begrenzt werden oder übernehmen die Transformation vom Target.

Eine besondere Form der Constraints stellt das Inverse-Kinematics-Constraint (IK-Constraint) dar. Die Constraints steuern die Objekte vom erst erstellten bis zum Letzt konstruierten Bone. Während die IK-Constraints die Bewegung ausführen. Die Constraints werden ausschließlich im Pose Mode erstellt (Beck 2015, S. 524ff.).

8.1 Die Kopf – Nacken – Bewegung

Bei der bisherigen Konstruktion wird die Kopfbewegung vom ganzen Körper beeinflusst. Das heißt, dass die Blickrichtung sich mit der Oberkörperausrichtung verändert. Um einen Erhalt der Blickrichtung nach vorne zu gewährleisten, werden folgende Schritte ausgeführt: Zunächst wird im Edit Mode die Parent-Child-Verbindung zwischen Skull- und Neck-Bone aufgehoben. Dazu wird der Skull-Bone selektiert (☞), mittels **[Alt] + [P]** und Clear Parent wird die Verbindung aufgehoben. Dabei bleibt die Parent-Child-Verbindung zwischen Neck- und Ribcage-Bone bestehen. Anschließend wird im Pose Mode der Skull-Bone selektiert (☞) und im Properties Editor im Icon Bone Constraints unter Add Bone Constraint den Constraint Copy Location hinzugefügt. Im Feld Target das Armature auswählen, unter Bone den Bone auswählen so wie die Eigenschaft, hier Neck-Bone, die der Skull-Bone übernehmen soll. Die Software Blender positioniert automatisch den Constraint-Bone am Head des zu folgenden Bones. Mit dem Schieberegler Head/Tail wird entschieden, an welcher Stelle die Position des Neck-Bones übernommen werden soll. Die Position von dem Tail des Neck-Bones soll übernommen werden, demgemäß wird der Wert vom Head/Tail auf 1 gesetzt (Abb. 19).

Der Kopf übernimmt somit die Position des Nackens und kann separat bewegt werden. Mit dem Hinzufügen dieses Constraints ändert der Skull-Bone seine Farbe in hellblau⁸.

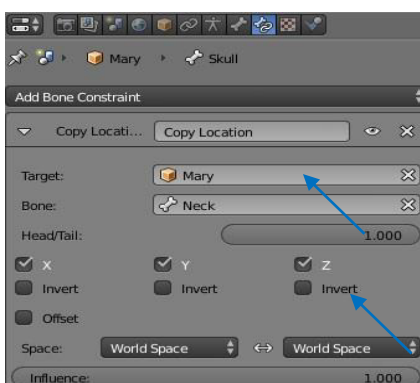


Abb. 19: Erstellung des Copy Rotation – Constraints

⁸ Blender definiert verschiedene Farben für unterschiedliche Funktionen der Bones. So ist es zu erkennen, welche Bones welchen Aufgaben unterliegen.

8.2 Die Armbewegung

Die Armbewegung⁹ erfolgt über einen unvollständigen IK-Constraint am MC-Bone. Im Pose Mode wird dazu der MC-Bone selektiert (☞). Anschließend im Properties Editor im Icon Bone Constraints unter Add Bone Constraint soll Inverse Kinematic ausgewählt werden. Unter Target und Pole Target werden keine Angaben gemacht. Es entsteht ein unvollständiger IK-Constraint. Mittels der inversen Kinematik steuert der IK-MC-Bone das Armature bis zum Ursprungs-Bone (hier zum Sacrum-Bone). Die sogenannte Kettenlänge der Armature Chain Length wird mit 4 definiert. Dementsprechend beträgt die Reichweite des IK-Constraints bis zum Humerus-Bone (MC-, Ossa carpi-, Radius- und Humerus-Bone). Es entsteht dabei eine braun-gestrichelte Linie zum Humerus (Abb. 20). Für eine Steuerung des Arms, wird der IK-Constraint im Pose Mode selektiert und bewegt. . Darüber hinaus kann die Position der Hand und des Armes über die jeweils einzelnen Bones individuell gesteuert werden.

Ein unvollständiger IK-Constraint erscheint im Properties Editor rot und der IK-Constraint-Bone am Mesh braun.

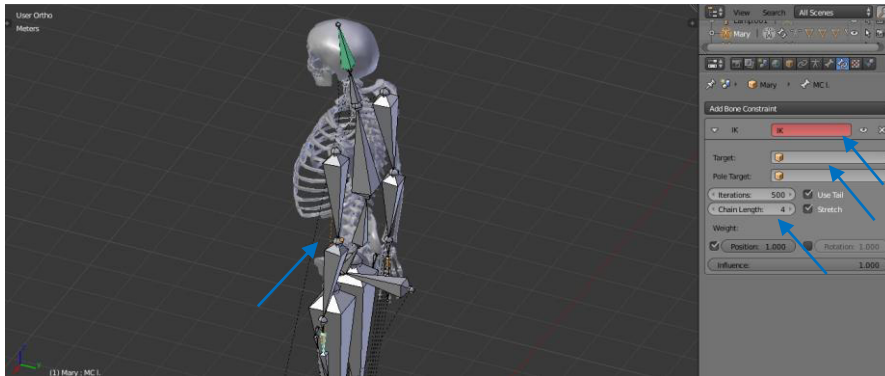


Abb. 20: unvollständiger IK-Constraint am MC-Bone

⁹ Das Hauptmerkmal dieser Arbeit liegt an den unteren Extremitäten. Daher wird hier die genaue Steuerung der Arme sowie Hände vernachlässigt.

8.3 Die Becken- und Oberkörperbewegung

Bei der bisherigen Konstruktion ist der Oberkörper nicht mit dem Unterkörper verbunden. Das Becken sowie der Oberkörper werden mittels einem Constraint am Sacrum in Abhängigkeit mit den unteren Extremitäten gebracht.

Als erstes wird ein Copy Location-Constraint für den Sacrum-Bone erzeugt. Dieser kann, ähnlich dem Constraint für die Kopfführung (Kapitel 8.1), manuell geschehen oder, je nach Belieben, über eine Tastaturkombination. Dazu werden nacheinander der CTRL-Hip- (Target) und der Sacrum-Bone selektiert und mit den Tasten **[Strg]** + **[Alt]** + **[C]** der Constraint Copy Location hinzugefügt (Abb. 21). Mit der erfolgten Tastaturkombination erstellt Blender automatisch einen Constraint für den Sacrum-Bone, so dass der manuelle Schritt über den Bone Constraints-Tab entfällt. In diesem Tab wird lediglich der Head/Tail-Schieber auf 1 gesetzt.

Damit folgen das Becken und der Oberkörper dem CTRL-Hip-Bone. Zudem steht der Oberkörper in einer Abhängigkeit mit den unteren Extremitäten.

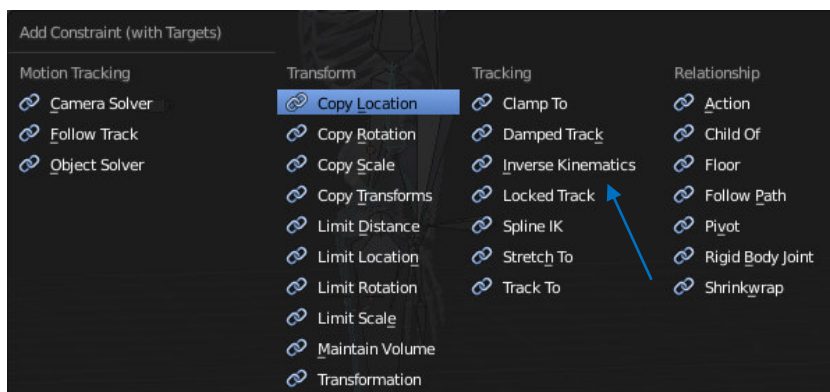


Abb. 21: Add Constraint (with Targets) – Fenster

8.4 Die Beinbewegung

Für die Steuerung der Beine werden zunächst die Tibie-Bones nacheinander mit IK-Constraints versehen. Das IK-Constraint wird durch den IK-Heel-Bone gesteuert. Eine Erzeugung des IK-Constraints erfolgt, im Gegensatz zur Erstellung der Armbewegung im Kapitel 8.2, mit einer Tastenkombination. Dazu werden im Pose Mode nacheinander der IK-Heel- und der Tibia-Bone selektiert, um anschließend mit der Tastenkombination **[⇧]** + **[I]** im Fenster Add IK To Active Bone (Abb. 23) automatisch

einen IK-Constraint zu erzeugen. Die Chain Length wird auf 1 gesetzt. Darüber wird die Bewegung der Tibia über den IK-Heel geführt (Abb. 24).



Abb. 22: Add IK – Fenster

Am Femur wird ein zweiter IK-Constraint, ausgehend vom CTRL-Knee-Bone, erstellt und die Chain Length ebenfalls auf 1 eingestellt (Abb. 23). Somit ist die Steuerungseigenschaft am Femur erstellt.

In Äquivalenz dazu, wird der oben beschriebene Vorgang am anderen Bein wiederholt.

Die Führung für die Beinsteuerung geht von zwei Bones aus: 1. Becken und 2. Ferse. Resultierend daraus, erfahren die Kniee eine Flexion. Über die beiden CTRL-Knee-Bones werden die Beine in die Neutral-Null-Stellung geschoben.

Bei dieser Konstruktion ist es möglich und später auch erforderlich, den Fuß separat zu bewegen. Der IK-Constraint der Tibia erlaubt das Einknicken des Beines invers bis zur Kniehöhe. Über die CTRL-Knee-Bones ist die Flexion der Beine zu animieren. Der CTRL-Hip-Bone dient der Höhenvariation des Skeletts dar.

Die IK-Constraints-Bones wechseln die Farbe zu Gelb, Copy Rotation zu Blau, die Chain Length wird in gelber, gestrichelter Linie dargestellt.

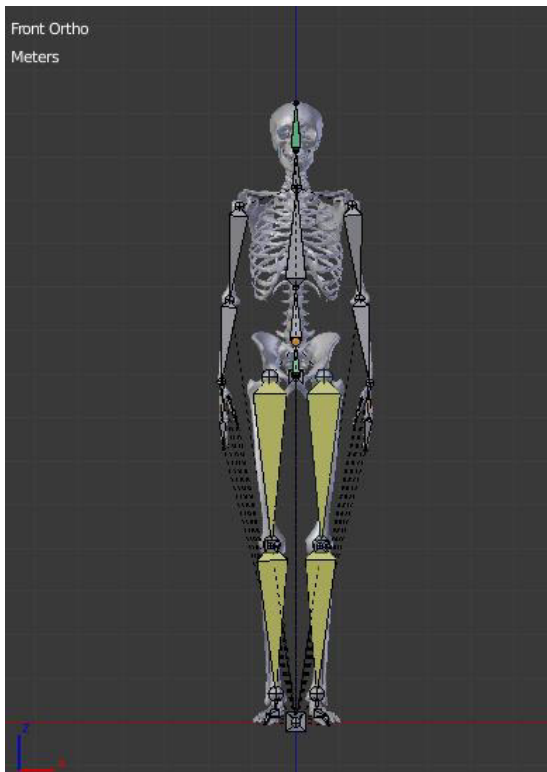


Abb. 23: IK-Constraints an Femora und Tibie

9 Die Ebenen

Blender bietet verschiedene Ebenen an, auf denen gearbeitet werden kann. Für eine bessere Übersicht können Bones auf unterschiedliche Ebenen aufgeteilt werden. Hier werden alle Bones in insgesamt vier Ebenen aufgegliedert: CTRL-Bones, Constraints, IK-Constraints sowie Hand_Upper Body_Leg_Foot. Die Bones, die auf eine andere Ebene verschoben werden sollen, werden im Pose Mode selektiert. Mit der Taste **[M]** öffnet sich das Fenster Change Bone Layers. Es wird die Ebene gewählt, wohin die Bones verschoben werden sollen (Abb. 24).

Im Properties Editor unter Data Object wird mit **[☑]** auf die verschiedenen Ebenen zugegriffen. Sollen alle Bones gleichzeitig am Mesh sichtbar werden, so können alle Ebenen parallel aktiviert werden, **[☑] + [☑]**. Bei der Verwendung von verschiedenen Ebenen, werden die Funktionen und Eigenschaften der Bones weder aufgehoben noch eingeschränkt.

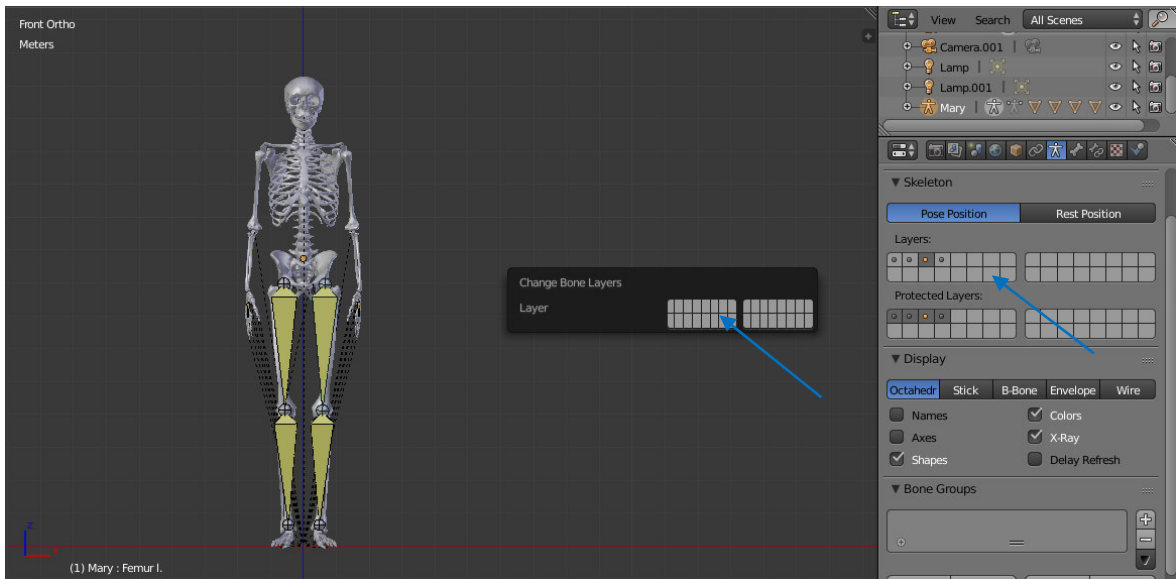


Abb. 24: Ebene mit IK-Constraints

10 Die Bone Groups

Zur weiteren Unterscheidung können Bones in sogenannte Bone Groups zusammen gefügt werden. Im Properties Editor unter Object Data im Panel Bone Groups werden die Bone-Gruppen erstellt. Zur Vereinfachung können die Bones aus den Ebenen zu einer Gruppe definiert werden. Im *Bone Groups* wird eine neue Gruppe mit dem Button + hinzugefügt. Mit einem Doppelklick auf den erscheinen Namen Group wird der Name umgeändert, in zum Beispiel CTRL-Bones. Es werden alle Bones, die durch diese Bone Group definiert werden sollen, selektiert und mit dem Button Assign beigefügt. Die CTRL-Bones befinden sich nun in einer gemeinsamen Gruppe. Diese kann weiterhin durch unterschiedliche Farben gestaltet werden. Unter dem Button Color Set werden diese ausgewählt, Beispielsweise rot für die Gruppe *Hand_Upper Body_Leg_Foot*, grün für *CTRL-Bone*, blau für *Constraint* und gelb für *IK-Constraint* (Abb. 25).

Die aktuelle Darstellung der Bones in Oktaeder-Form ist groß und versperrt die Sicht auf das Skelett. Blender bietet unter Data Object im Fenster Display verschiedene Bone-Typen an. Eine geeignete Form, die die Sicht nicht versperrt ist die sogenannte Stick-Form (Abb. 25). Die Bones ändern ihre Form in eine Linie mit Schnittpunkten als Zeichen vom Anfangs- und Endpunkt eines Bones.

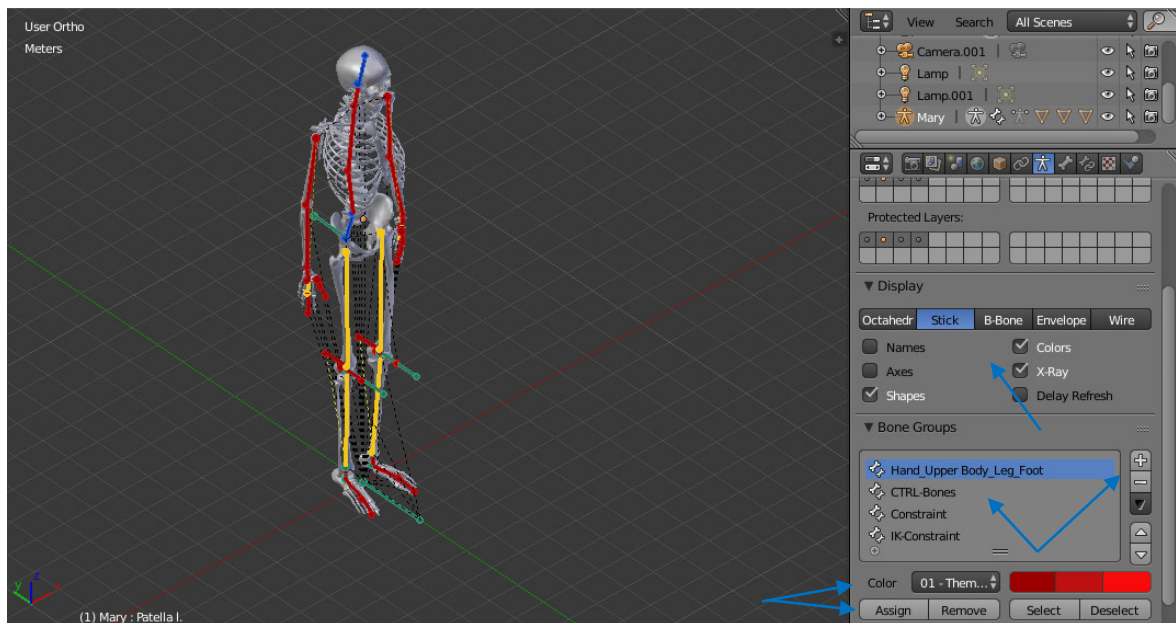


Abb. 25: Bones untergliedert nach Bones Groups, Farben und in Stick-Form

11 Das Custom Shape

Das Custom Shape stellt die Bones in einer weiteren Form dar. Die Bones werden nicht mehr als Ganzes (beispielsweise als Stick oder Oktaeder) verkörpert, sondern durch eine individuelle Kontur. Meistens wird dafür ein Kreis verwendet. Somit wird die Sicht auf das Mesh zusätzlich erweitert. Custom Shape dienen überdies der Vereinfachung der Steuerung von Bones (Beck 2015, S. 521f.).

Im Object Mode wird ein Kreis (Circle) erstellt. Der Cursor wird am Ursprungspunkt über die Tastenkombination $\square + \square$ positioniert. Anschließend wird mit $\square + \square$ ein Kreis mit beispielsweise 12 Ecken erzeugt¹⁰. Die Anzahl der Ecken wird im Tab Add Circle eingestellt (Abb. 26).

¹⁰ Alternativ kann der Kreis über TOOL SHELF im Tab *Create* erstellen werden.

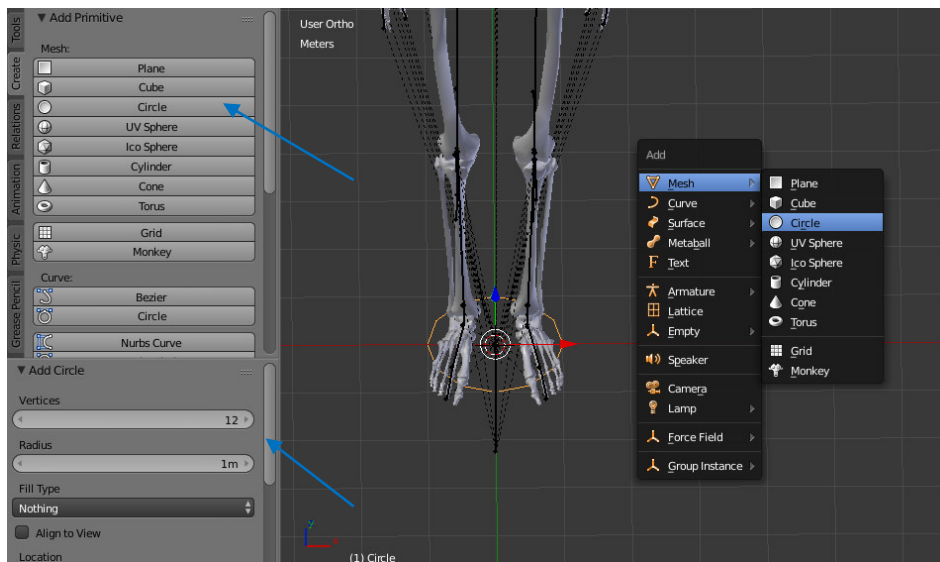


Abb. 26: Erstellung eines Kreises für Custom Shape Funktion

Die Custom Shape Funktion wird für die Hand- und Arm-Bones, für die Wirbelsäulen-Bones sowie für die Femora- und Tibie-Bones erstellt. Zugleich bleiben der MC- und der Sacrum-Bone von der Custom Shape Funktion ausgeschlossen. Die Ausführung erfolgt folgendermaßen: Den Humerus-Bone im Pose Mode selektieren. Im Properties Editor im Icon Bone unter Display im Fester Custom Shape Circle auswählen und Wireframe aktivieren (Abb. 28). Der Humerus-Bone wird in Form von einem Kreis sichtbar (Bone-Shape). Die Ausrichtung der Kontur erfolgt im Edit Mode mithilfe von Transform manipulator Rotate, indem der Custom Shape zuvor im Object Mode selektiert wird.

Im Object Mode wird im Anschluss der Custom Shape im Properties Editor im Icon Object in beispielsweise "WGT_Arms" umbenannt (Abb. 27). Das WGT steht für Widget, welches eine Zusammensetzung aus den Wörtern Windows und Gadget ist. Die oben genannten Arm- sowie Hand-Bones werden zu diesem Custom Shape hinzugefügt, indem WGT_Arms nacheinander den einzelnen selektierten Bones im Icon Bone zugeteilt wird.

Zwei weitere Custom Shapes werden jeweils für die Spine-Bones sowie für Femora- und Tibie-Bones erstellt (Abb. 28).

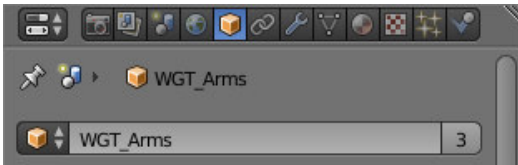


Abb. 27: Properties Editor: Icon Object

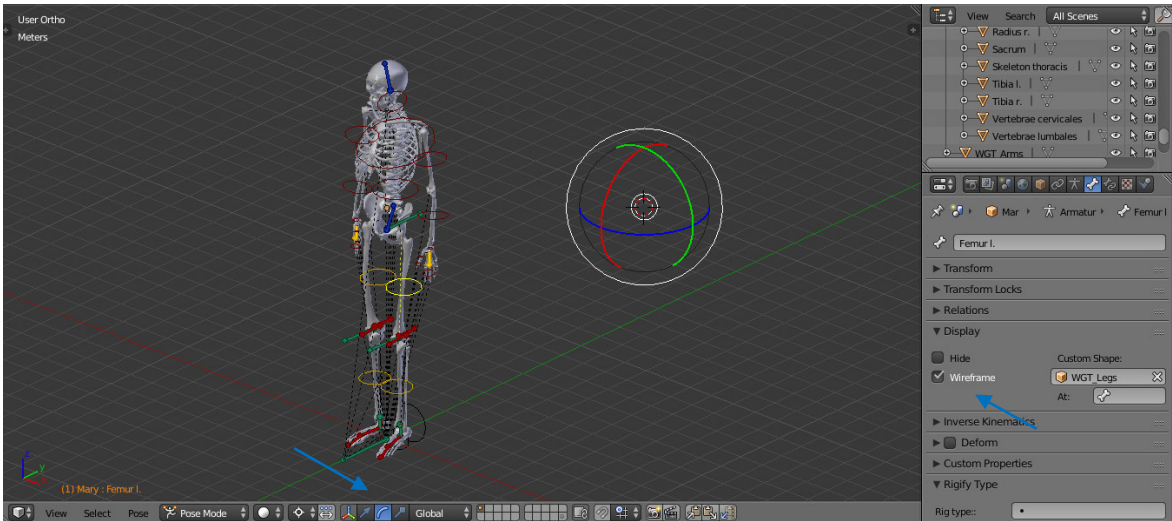


Abb. 28: Custom Shapse am Mesh

Der Haupt-CTRL-Bone bekommt einen eigenen Custom Shape. Dieser wird auf einer anderen Ebene erstellt, die im Object Mode im Window Header gewählt wird (Abb. 29).

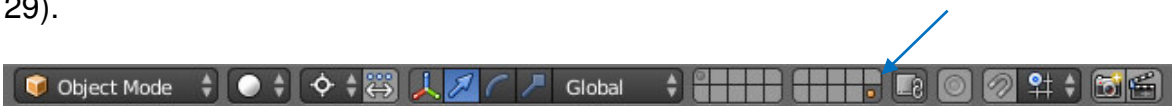


Abb. 29: Ebene für den Custom Shape für den Haupt-CTRL-Bone

Ein regelmäßiges Achteck wird im Ursprung mit der Tastenkombination ($\text{Ctrl} + \text{C}$, $\text{Ctrl} + \text{A}$) erzeugt. Im Edit Mode im Window Header erfolgen die Punktselektion sowie eine Erstellung einer viereckigen Fläche ($\text{Ctrl} + \text{A}$, "Plane"). Diese wird zum Kreisinneren skaliert (S , Abb. 30). Mit Entf + Only Faces im Delete-Fenster bleiben nur die Umrisse der Fläche über.

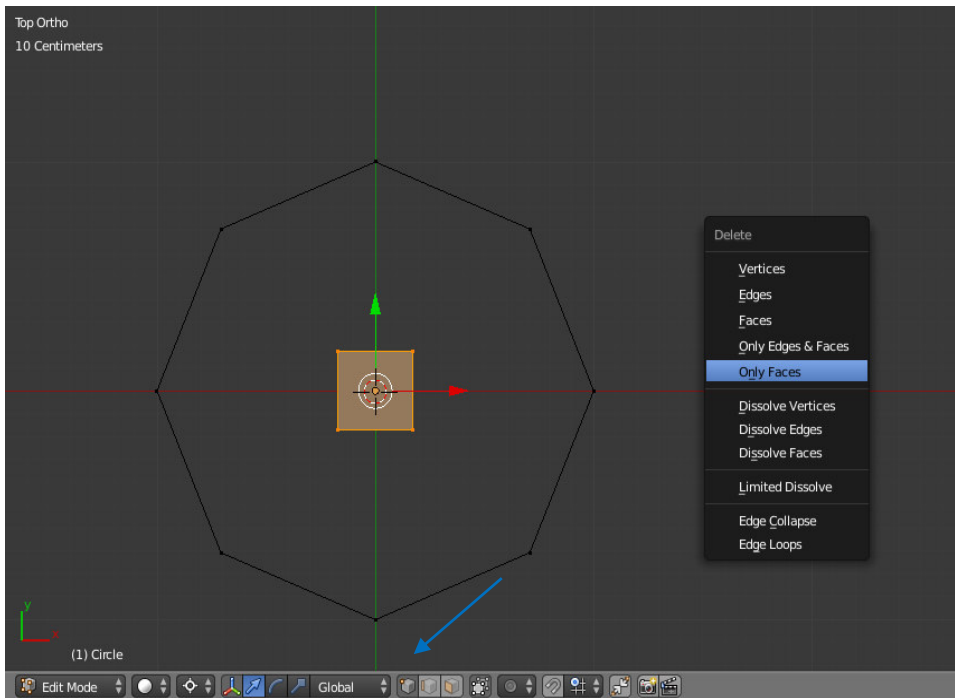


Abb. 30: I Hauptkreiserstellung des Custom Shapes für den Haupt-CTRL-Bone

Mit der Punktselektion (Abb.30) werden die Ecken der Fläche ausgewählt (**Alt** + **Q**) und zum Rand des Kreises positioniert (beispielsweise mit **Q** + x-Koordinate). Die zwei Außenpunkte der Fläche selektieren und im 3D Cursor Median Point auswählen. Die Punkte in die x-Richtung extrudieren (**E** + **X**) und anschließend auf null skalieren (**S** + **0**). Mit **Alt** + **M** im Fenster Merge At Center auswählen (Abb. 31). Mit dieser Funktionen werden die beiden Schrägen in einem Punkt vereint.

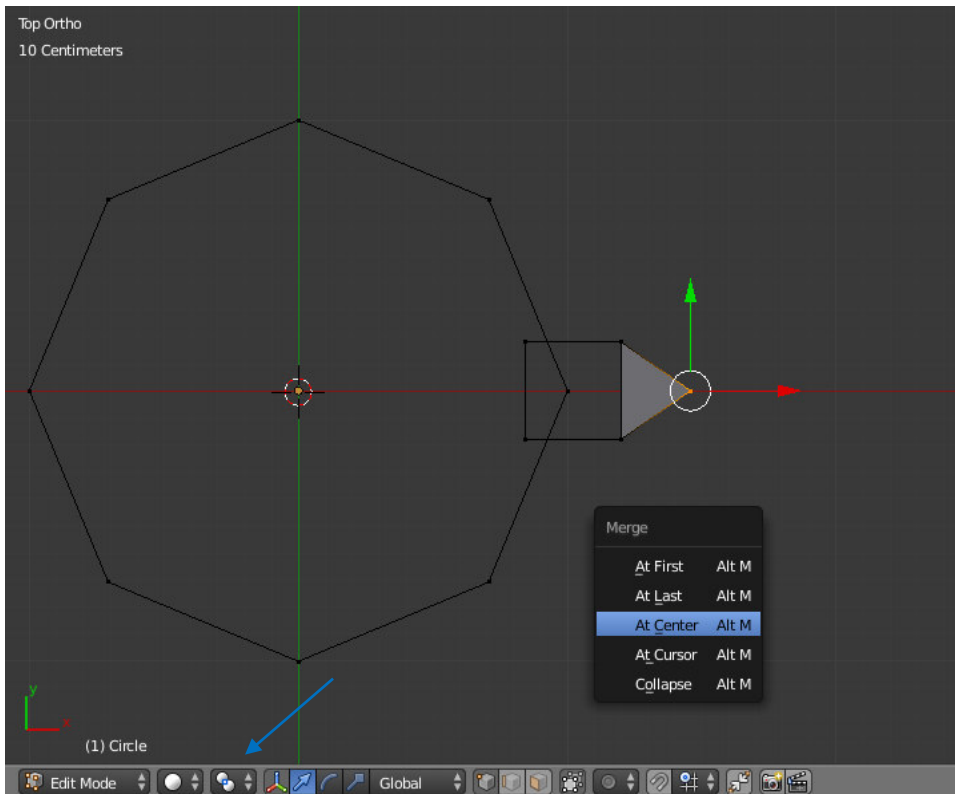


Abb. 31: II Hauptkreiserstellung des Custom Shapes für den Haupt-CTRL-Bone

Die drei Eckpunkte vom entstandenen Dreieck werden selektiert sowie mit **[Entf]** + Only Faces die gefüllte Fläche entfernt. Mit dem Mauscursor auf einen beliebigen Punkt der neuen Kontur zeigen, mit Taste **[L]** wird diese als Ganzes selektiert. Anschließend den 3D-Cursor am Pivot Point positionieren, um die erschaffene Form viermal zu duplizieren und an jeden zweiten Punkt des Kreises positionieren (**[⇧]** + **[D]** + **[R]** + **[9]** **[0]**). Die verbliebenen Verbindungslinien in der Kontur werden selektiert (**[Alt]** + **[⇧]**) und mit **[Entf]** + Edges im Delete-Fenster gelöscht. Im Outliner wird das Profil umbenannt, beispielsweise in WGT_Root (Abb. 32).

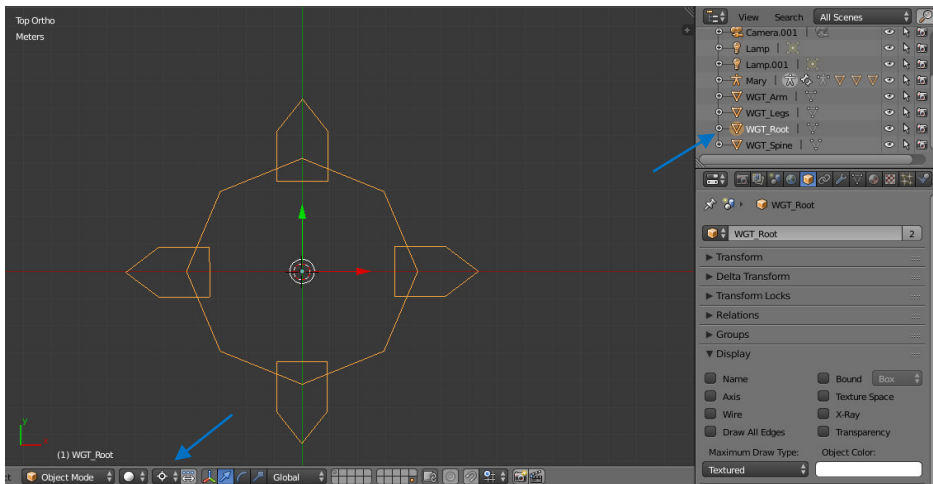


Abb. 32: fertig erstelltes Custom Shape für den Haupt-CTRL-Bone

Dieses Custom Shape für den Haupt-CTRL-Bone wird auf seiner entsprechenden Ebene zugewiesen. Im Pose Mode wird der Haupt-CTRL-Bone selektiert, im Properties Editor im Icon Bone unter Display im Fester Custom Shape WGT_Root auswählen und Wireframe aktivieren. Der Custom Shape wird am Boden angezeigt (Abb. 33).

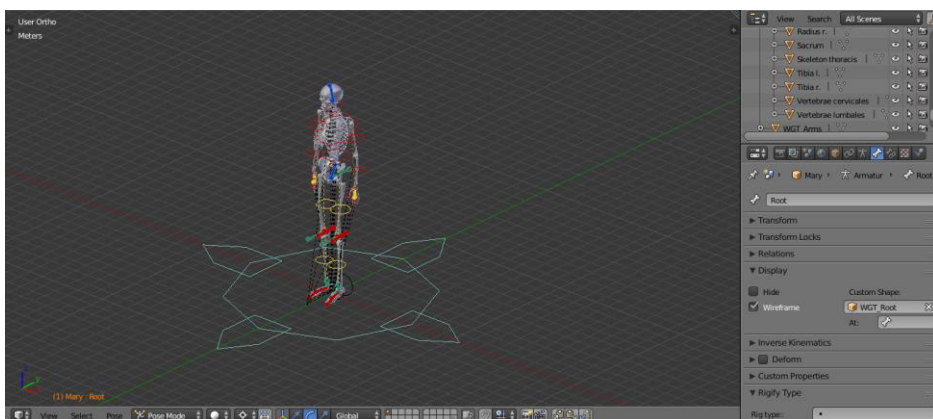


Abb. 33: Mesh mit Custom Shapes

Mit dem Root-Custom Shape wird das Mesh bewegt. Damit das Mesh auch die Rotationsbewegungen des Custom Shapes übernimmt, wird eine Keep Offset-Verbindung zwischen Skull- sowie Sacrum-Bone und dem Haupt-CTRL-Bone erstellt. Im Outliner OUTLINER unter Armature sowie Pose im Bone Root nacheinander Skull, Sacrum und Root selektieren ($\text{Ctrl} + \text{A}$) und im Edit Mode eine Parent-Child-Verbindung erstellen ($\text{Strg} + \text{P}$) und Keep Offset).

12 Animation des Bewegungsablaufs

Animation stammt aus dem lateinischen von *animare*, "zum Leben erwecken". In der Technik entsteht ein bewegtes Bild in der schnellen Abfolge von Einzelbildern. (Beck 2015, S. 448).

In Blender werden dem Rig zu bestimmten Zeitpunkten Schlüsselpositionen (Keyframes) zugewiesen. Ein Keyframe enthält zum einen einen definierten Zeitpunkt, zum anderen Parameter, wie beispielsweise Rotation oder Position, des Objekts. Zwischen den gesetzten Keyframes werden die Daten durch Blender mathematisch interpoliert (Wartmann 2011, S. 217).

Der Animationsvorgang wird im Pose Mode durchgeführt. Vor Beginn werden einzelne Voreinstellungen vorgenommen. Im Screenshot Layout wird vom Default in Animation gewechselt. Im Fenster neben dem Screenshot Layout wird Scene mit einem Doppelklick in beispielsweise world umbenannt (Abb. 34).

Im Time Line Editor den Aufnahme Icon aktivieren, im Feld (Schlüsselsymbol) daneben das Keying Set LocRotScale auswählen (Abb. 34). Dadurch speichert Blender automatisch vorgenommene Änderungen (Location / Rotation / Scale) am Mesh.

Die Animation eines Gangzyklus erfordert viele Keyframes. Der Action Editor (Abb.34), der im Dope Sheet Editor zu finden ist, vereinigt die große Anzahl der Keyframes zu einer logischen Gruppe zusammen (Beck 2015, S. 461f.; 475f.).

Eine neue Aktion wird im Action Editor mit dem +-Button erstellt und betitelt (Abb. 34). Insgesamt werden vier Aktionen nacheinander zu den zu animierenden Zeitpunkten erzeugt: stand, walk, crouch down sowie one leg stand. Dabei wird jedes Mal die Fake-User (F) Schaltfläche aktiviert.

Um einen realitätsnahen Bewegungsverlauf zu erreichen, wird im Dope Sheet Editor unter Key die Interpolationsmethode Bezier eingestellt. Diese Methode erzeugt einen Eindruck von Trägheit von Massen.

Für die während der Animation präzisere Positionierung der unteren Extremitäten sowie der Winkelberechnungen werden von den unteren Extremitäten die Custom Shapes rausgenommen, indem einzeln bei jedem zugehörigen Bone im Properties Editor im Icon Bone der Custom Shape Custom Shape entfernt werden (Abb.34).

Beim Gehen soll der Kopf in die Gangrichtung zeigen. Daher wird beim Skull-Bone im Icon Bone unter Relation inherit rotation eingestellt. Somit übernimmt der Skull-Bone die Rotation vom Eltern-Objekt, dem Root-Bone, und schaut in dessen Richtung, der Gangrichtung.

Im Pose Mode wird das Mesh mit der Tastenkombination **Strg** + **Pad3** in die Seitenansicht gebracht.

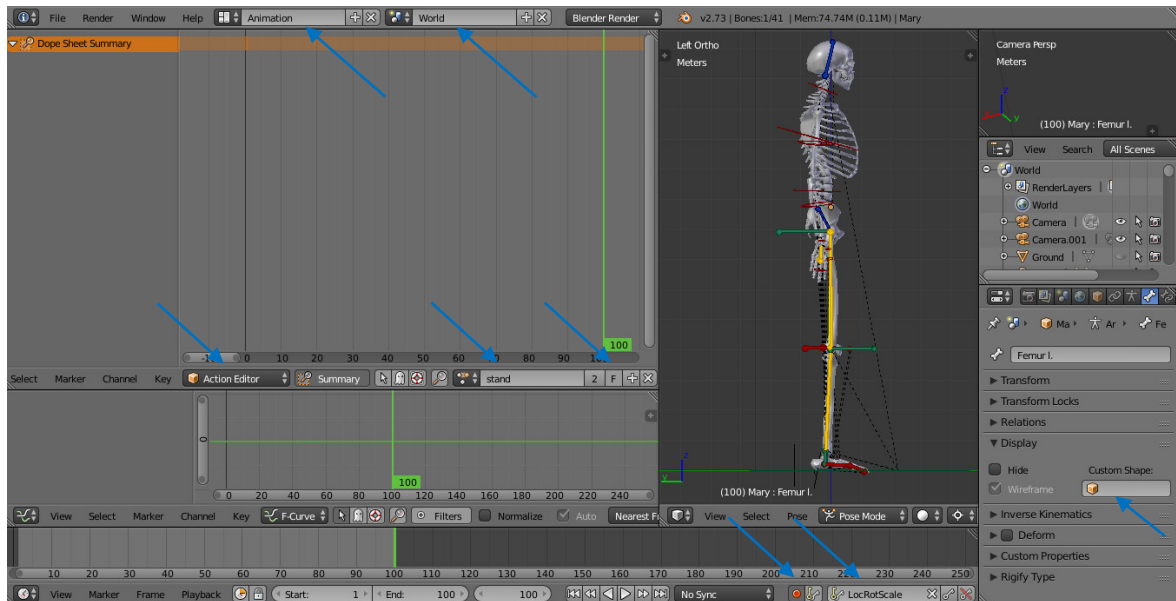



Abb. 34: Seitenansicht; Voreinstellung für die Animation

In den folgenden Schritten wird erörtert, wie das Skelett in die Standposition animiert wird. Der Schieberegler (Playhead) wird vorab mit **11** auf Frame 1 platziert. Das Mesh wird mit der Taste **A** selektiert und mittels der Taste **I** werden die Keyframes an Frame 1 gesetzt. Das Skelett befindet sich in einer Neutral-Null-Stellung. Zur genauen Überprüfung der Winkel¹² in der Neutral-Null-Stellung werden von den entsprechenden Bones die Koordinaten im Properties abgelesen und daraus die Winkel bestimmt. Gegebenenfalls werden die Positionen der Bones nachjustiert. Alle Winkelangaben sowie Bestimmung der Winkelgrade sind im Anhang Phasenwinkel dargelegt. Die Ausrichtung erfolgt ausschließlich über den Transformation

¹¹ Playhead kann alternativ über die Pfeiltasten gesteuert werden.

¹² Das Hauptmerkmal der Winkelausrichtung liegt in dieser Arbeit auf den unteren Extremitäten, dementsprechend auf dem Hüftgelenk, dem Kniegelenk sowie dem Sprunggelenk.

Manipulator Translate oder die Tastatureingabe **R** + **X** / **Y** / **Z**! Dabei werden die betreffenden Bones über die Maus  markiert. Überdies kann zwischen den globalen und den lokalen Koordinaten gewechselt werden.

Ein Wechsel der Koordinatenachsen führt häufig zu einer Erleichterung von Animationsvorgängen. Anschließend wird der Playhead auf Frame 41 platziert und die Keyframes werden erneut mit **I** abgespeichert (Abb. 35).

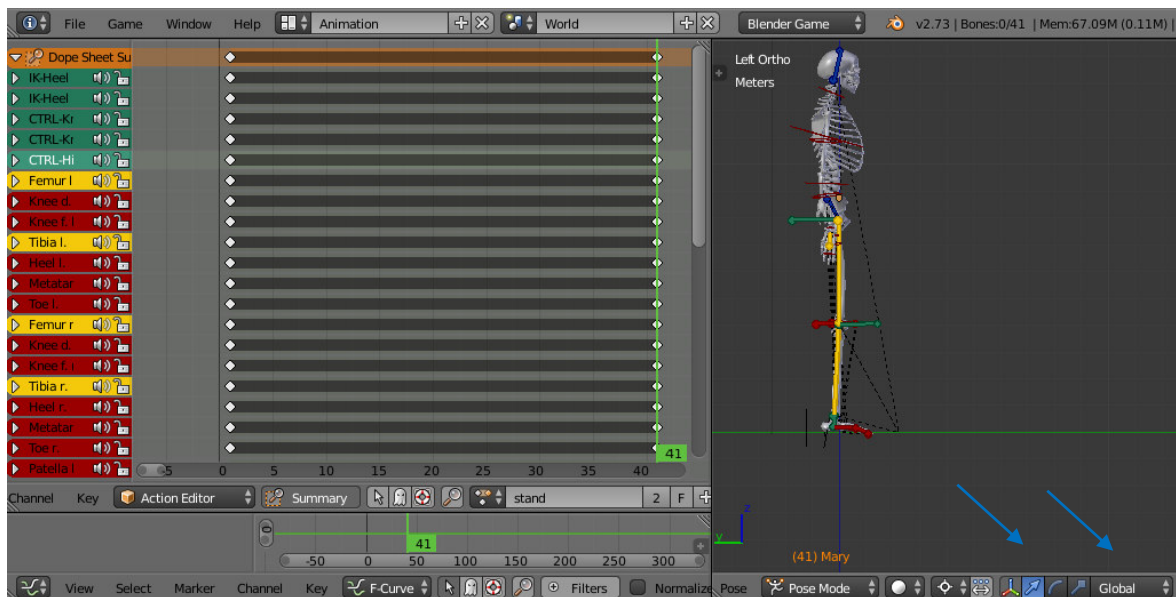



Abb. 35: Animation, stand

Nach der erfolgten Standanimation, folgt die Animation des Gangzyklus. Dazu wird eine neue Aktion mit dem +-Button im Action Editor erzeugt und als walk definiert. Eine Zeitspanne von insgesamt 39 Frames wird für einen Gangverlauf festgelegt. Der äußerste, obere Keyframe wird mit der Maus  ausgewählt und mit der Taste **Entf** entfernt. Die Keyframes an Frame 1 bleiben bestehen.

Der Fokus in diesem Gangzyklus liegt auf dem rechten Bein, dem sogenannten Referenzbein. An diesem Bein werden die Winkel, wie im Anhang Phasenwinkel dargestellt, bestimmt. Das kontralaterale Bein, sowie Becken, Rumpf, Kopf und obere Extremitäten werden per Augenmaß positioniert. Es empfiehlt sich die unteren Extremitäten sowie das Becken parallel zum Referenzbein zu positionieren.

Das Skelett wird an Frame 1 in die erste Gangphase¹³, dem sogenannten Initial contact, gesetzt. Die Realisierung der einzelnen Gangphasen erfolgt allein, wie oben beschrieben, über den Transformation Manipulator Translate oder die Tastatur. Befindet sich das Mesh im Initial contact, werden im Anschluss die Winkel geprüft und gegebenenfalls nachjustiert. Mit der Taste **I** werden die Keyframes anschließend abgespeichert (Abb. 36).

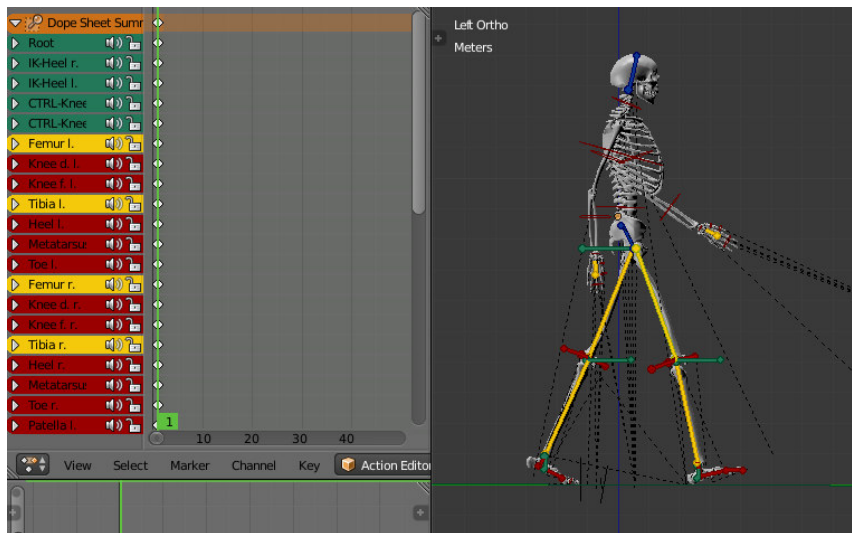


Abb. 36: Animation walk, Frame 1, Initial contact

Der Playhead wird auf Frame 4 positioniert und die nächste Gangphase, Loading response, wird wie oben beschrieben animiert und abgespeichert (Abb. 37). Dieser Vorgang wird bei jeder Gangphase wiederholt. Die Gangphasen werden wie in Tabelle 4 gesetzt. Darüber hinaus wird die Mid stance unterteilt (Götz-Neumann 2011, S. 81) und an Frame 10 (ca. die Mitte dieser Gangphase) erstellt.

¹³ Der Gangzyklus wird in insgesamt 8 Gangphasen unterteilt (Götz-Neumann 2011, S. 12f.). Die Gangphasen sind in der unten genannten Tabelle 4 aufgelistet.

<i>Gangphase</i>	<i>Frame</i>	<i>Abbildung</i>
Initial contact	1	36
Loading response	4	37
Mid stance		
mid Mid stance	10	38
late Mid stance	16	39
Terminal stance	21	40
Pre-swing	25	41
Initial swing	31	42
Mid swing	35	43
Terminal swing	38	44

Tabelle 4: Einteilung der Gangphasen zu den entsprechenden Frames



Abb. 37: Animation walk, Frame 4, Loading response

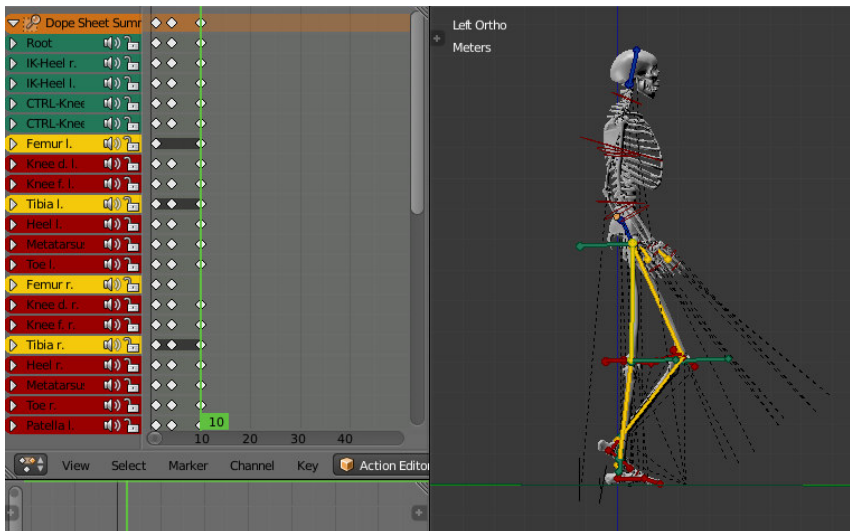


Abb. 38: Animation walk, Frame 10, mid Mid stance

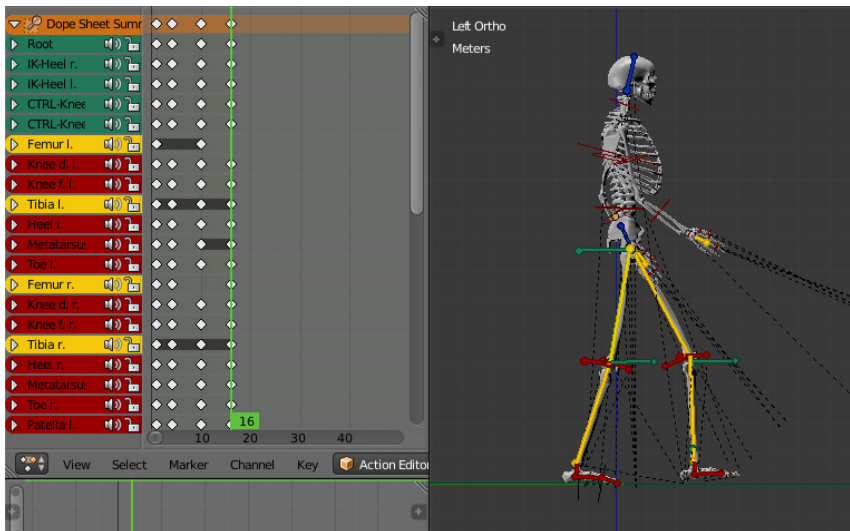


Abb. 39: Animation walk, Frame 16, late Mid stance

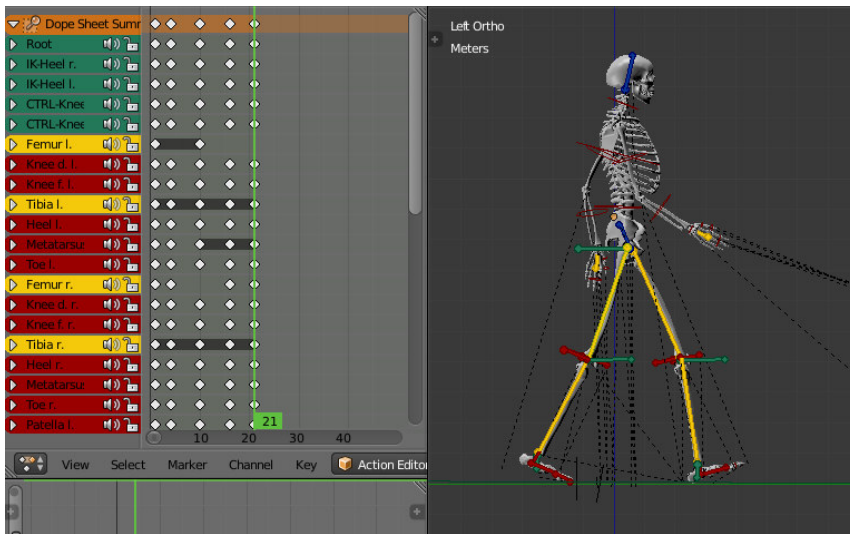


Abb. 40: Animation walk, Frame 21, Terminal stance

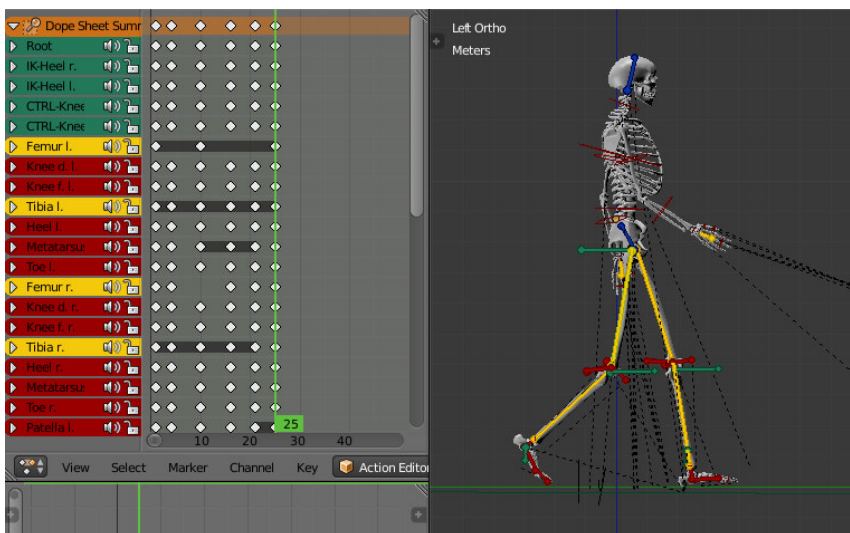


Abb. 41: Animation walk, Frame 25, Pre-Swing

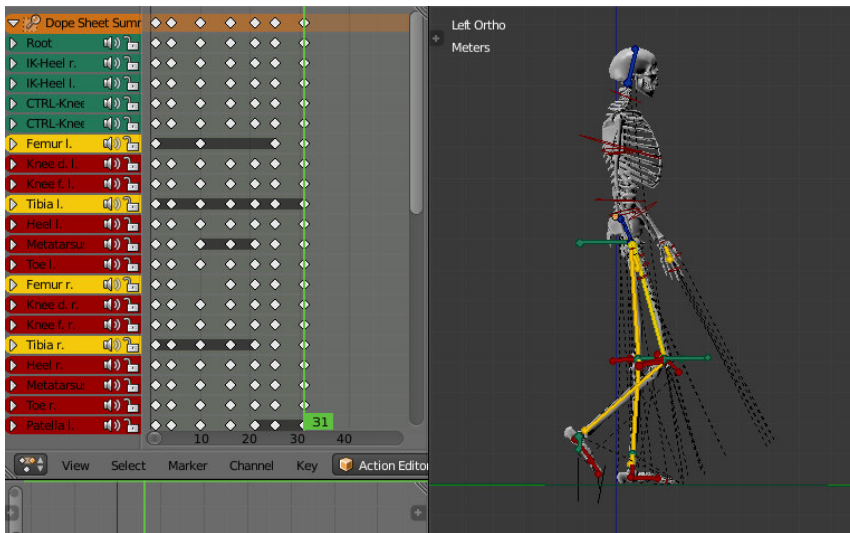


Abb. 42: Animation walk, Frame 31, Initial swing

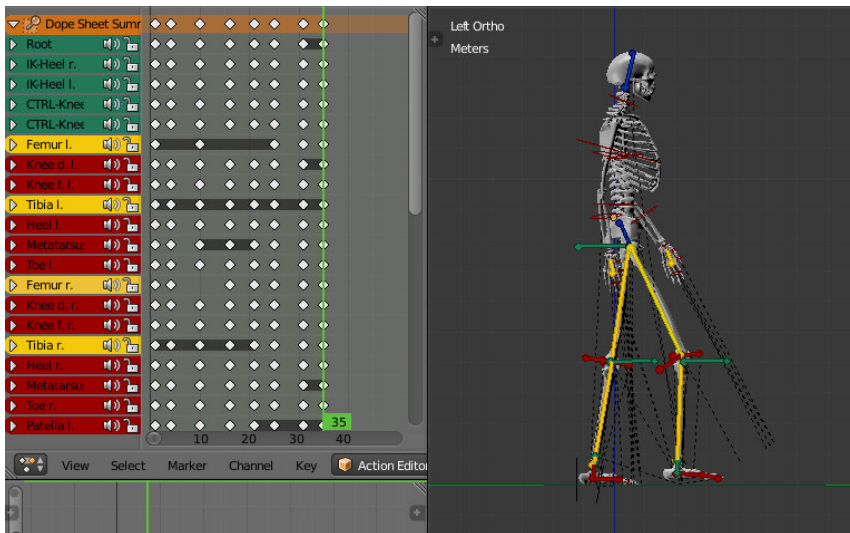


Abb. 43: Animation walk, Frame 35, Mid swing

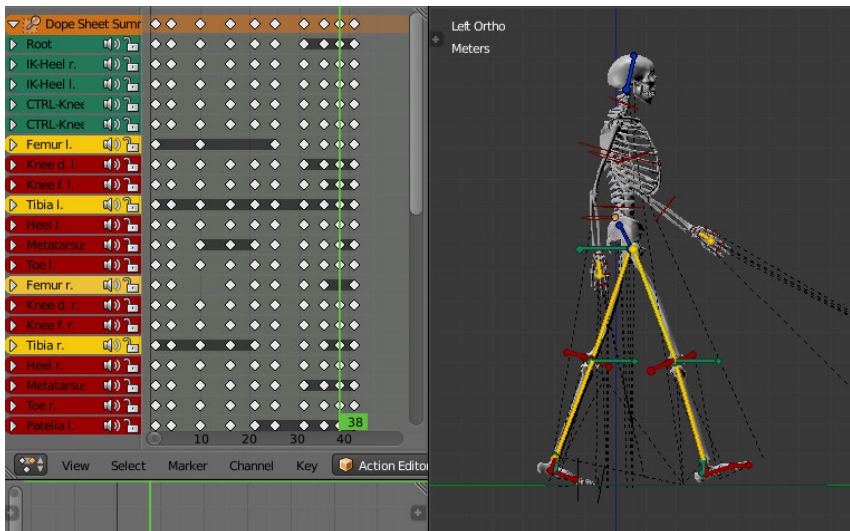



Abb. 44: Animation walk, Frame 38, Terminal swing

Im Anschluss an die Animation der Gangphasen wird Frame 1 auf Frame 41 kopiert: Den obersten Keyframe von Frame 1 mit der Maus  selektieren, mit der Tastenkombination **[Strg]+[C]** kopieren und auf Frame 41 mittels **[Strg]+[V]** einfügen (Abb. 45). Auf diese Weise entsteht ein flüssiger Übergang in der Animationsschleife.

Im Timeline Editor wird die Animationszeit kontrolliert und koordiniert. An dieser Stelle erfolgt eine Definition des Beginns bei Frame 1 und ein Ende bei Frame 40. Mittels der Abspielkontrollen kann die Animation gestartet und beendet¹⁴ werden. Es entsteht eine Bewegungsschleife, die nach dem 40sten Frame automatisch erneut startet.

Bei laufender Animation wird so die gleichmäßige Bewegung des Skeletts überprüft. Bei Bedarf wird die Bewegung zwischen den gesetzten Frames nachjustiert. Hierbei müssen die eingefügten Keyframes nicht mehr über die Taste **[I]** gespeichert werden. Das ist darin begründet, dass der voreingestellte Keying Set LocRotScale die Veränderungen der Animation automatisch abspeichert.

¹⁴ Alternativ kann das Abspielen der Animation über **[Alt]+[A]** erfolgen.

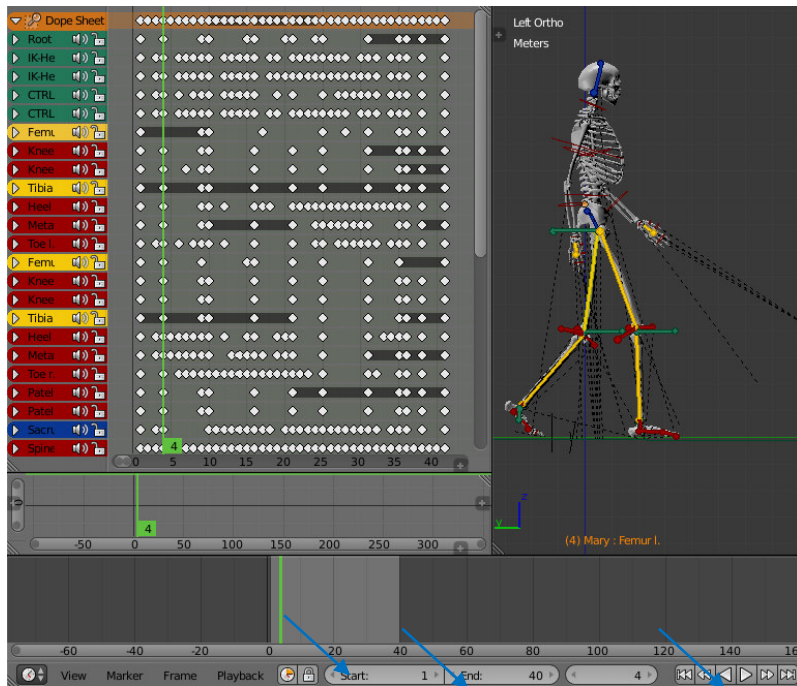


Abb. 45: vollendete Animation des Gangzyklus
 Am Beispiel von Frame 4, Loading response

Entsprechend der oberen Beschreibung werden die Animationsvorgänge one leg stand und to crouch down positioniert (Abb. 46; 47). Für die beiden Animationen wird eine Zeitspanne von jeweils 90 Frames festgelegt.

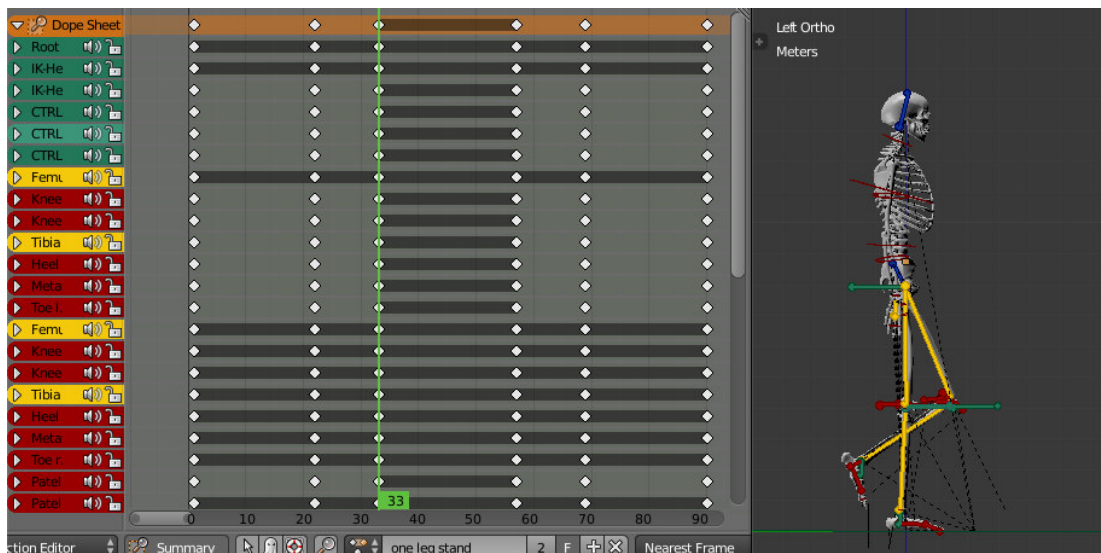


Abb. 46: Animation one leg stand, Frame 33

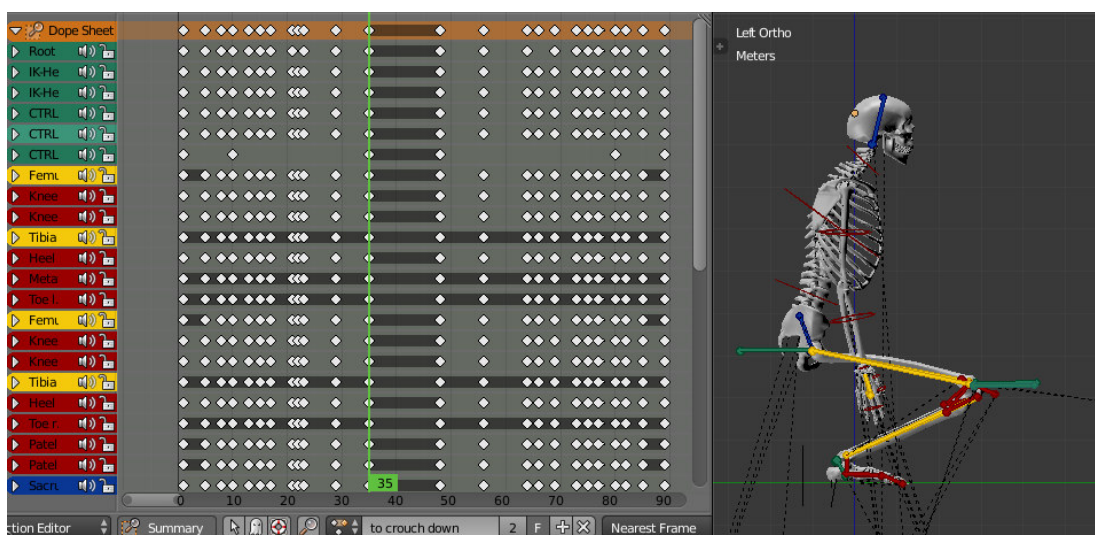


Abb. 47: Animation to crouch down, Frame 35

Der Gangzyklus wird ebenfalls in der slow motion-Animation dargestellt (Abb. 48). Hier werden die Keyframes aus walk kopiert und in die neu erstellte Aktion slow motion eingefügt. Anschließend mit der Taste **A** werden die Keyframes selektiert und über den Playhead auf Frame 1 gesetzt. Mittels **S** werden die Keyframes bis Frame 120 skaliert. Bei laufender Animation wird die fließende Bewegung des Skeletts noch einmal überprüft und gegebenenfalls nachjustiert.

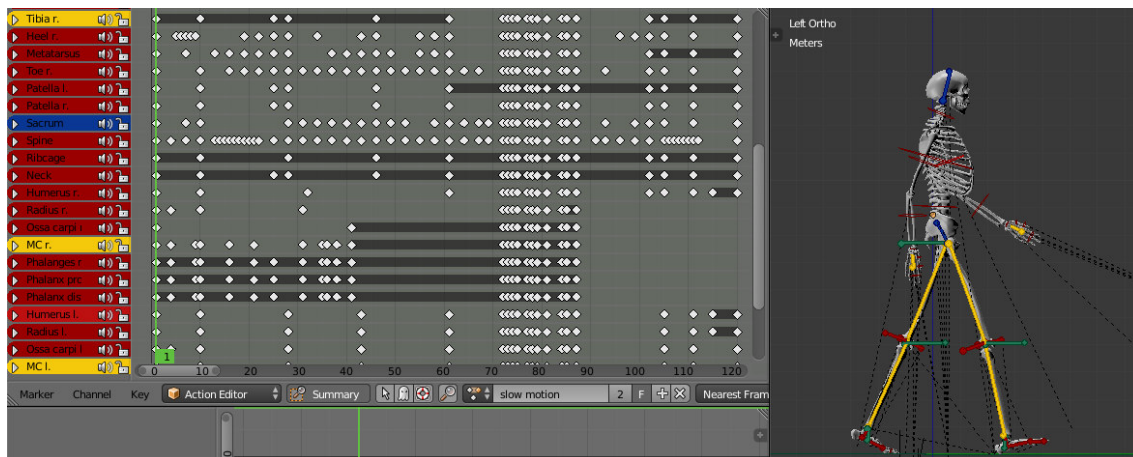


Abb. 48: Animation Frame 1, slow motion

Für die spätere Animation der Kraftanzeigen am Hüft- und Kniegelenk (Kap. 18) wird im Timeline Line Editor der Aufnahme Icon deaktiviert sowie das Keying Set LocRotScale entfernt.

Die an den unteren Extremitäten entfernten Custom shapes werden wieder eingefügt (im Properties Editor unter Bone). Ergänzend werden die Bones an den Zehen als Custom Shape deklariert. Zusätzlich können die Bones in Form von Wire anstatt von Stick dargestellt werden. Das hat den Vorteil, dass die Ansicht dieser Bones weniger Platz in Anspruch nimmt, als die Stick-Darstellung.

13 Hintergrund

Der Hintergrund der Szene world wird individuell angepasst. Dieser wird im Properties Editor im Icon World reguliert. In den Fenstern Horizont Color, Zenith Color und Ambient Color werden die Farbabstufungen für den Hintergrund bestimmt, hier hellblau (Abb. 49). Die Farbverläufe werden über Paper Sky, Blend Sky und Real Sky eingestellt. Im Properties unter Display werden die Einstellungen mit World Background aktiviert (Abb. 49; 50).

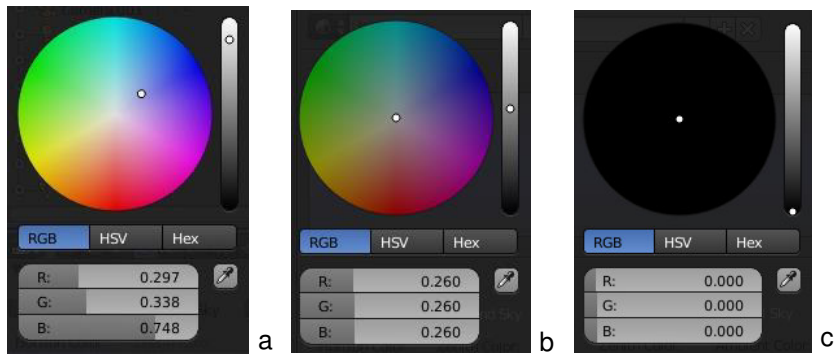


Abb. 49: Farbabstufung für den Hintergrund
a) Horizont, b) Zenith color, c) Ambient

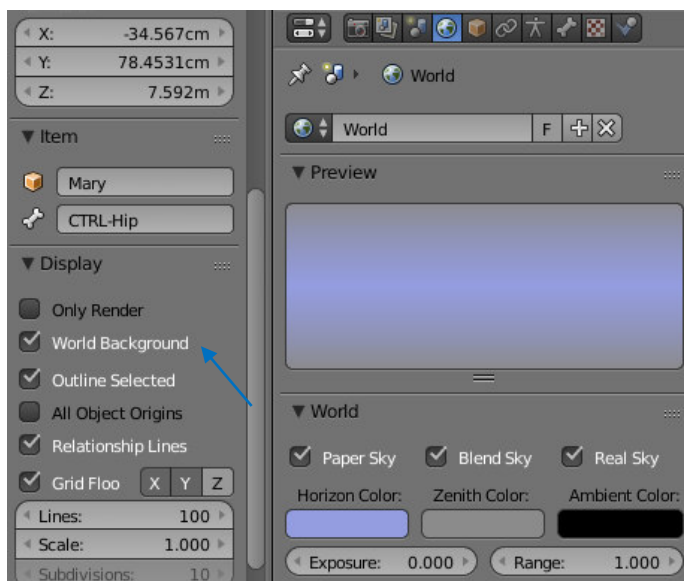


Abb. 50: Aktivierung des World Background

14 Untergrund Game Engine

Das Skelett braucht für die Game Engine (Kap. 20) einen virtuellen Untergrund, auf dem es laufen kann. Ohne diesen virtuellen Untergrund, wäre eine Simulation einer Ganganimation nicht nachahmbar. Für die Umsetzung wird der Cursor im Object Mode in den Ursprungspunkt gebracht. Die Shortcuts lauten für die Umsetzung: $\square + S$, mit $Alt + A$ eine Fläche (Plane) erzeugen. Im Properties wird unter Shading GLSL eingestellt und im Item die Bezeichnung der erzeugten Fläche in Ground geändert. Im Pose Mode wird in Item Material mit + ein neues Material erzeugt und liso eingestellt (Abb. 51).

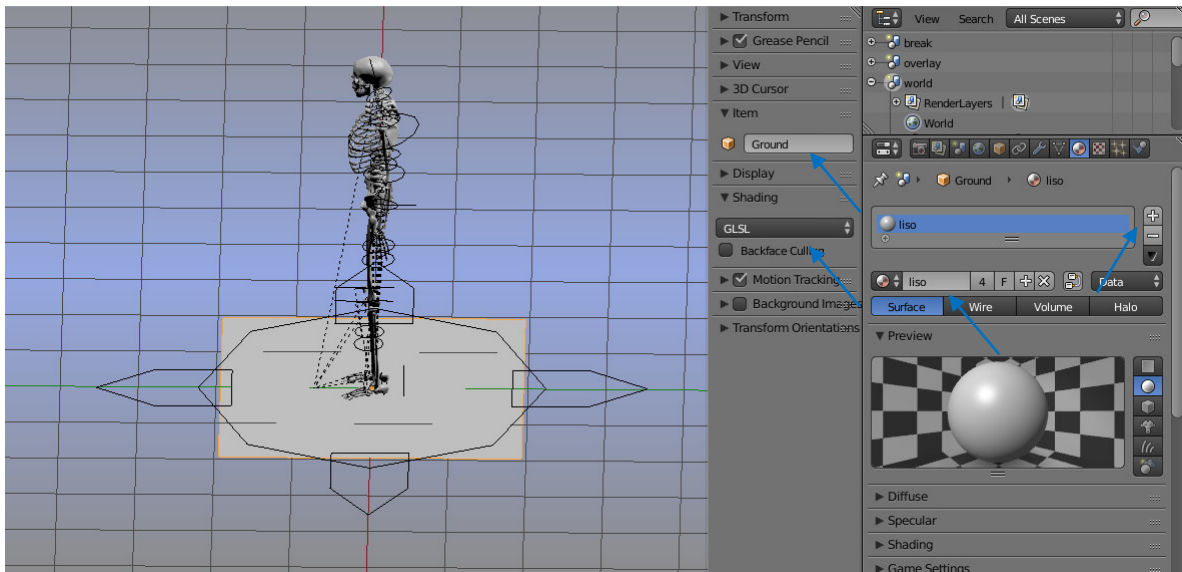



Abb. 51: Erstellung des Virtuellen Bodens

15 Kamera und Licht

Im Screen Layout wird von Animation zu Default gewechselt. Im Viewport wird das Material eingestellt. Die Kameras sowie die Lichtquellen werden im Object Mode eingefügt. Zur vereinfachten Kamera- sowie Lichtausrichtung können die von Blender von Beginn an vorhandene Kamera und Sonne gelöscht werden (, **Entf**).

15.1 Kamera

Zunächst wird eine Kamera am Pivot Point erschaffen. Der Cursor wird dafür am Ursprungspunkt mit den Shortcuts **[Alt] + [S]**, Cursor to center platziert. Im Tool shelf wird unter Create Camera ausgewählt. Durch diese Kamera soll das Skelett als Ganzes betrachtet werden. Demgemäß wird diese Kamera in Item als full body cam deklariert. Die Kamera wird vor dem Mesh positioniert und gegebenenfalls etwas größer skaliert (**[S]**). Mit der Taste **[Pad 0]** wird die Ansicht in die Kameraperspektive gewechselt. Aus dieser Ansicht wird das Skelett im Bild zentriert und kann mithilfe des Transformation manipulators nachjustiert werden. Im Properties Editor im Icon Object Data wird unter Depth of Field der Bereich der Schärfentiefe eingestellt. Im Fenster Focus wird das Objekt ausgewählt, auf dessen Zentrum die

Kameraschärfe fokussiert wird. Hier soll Kamera auf das Skelett fokussiert werden (Abb. 52; 53) (Beck 2015, S. 275).

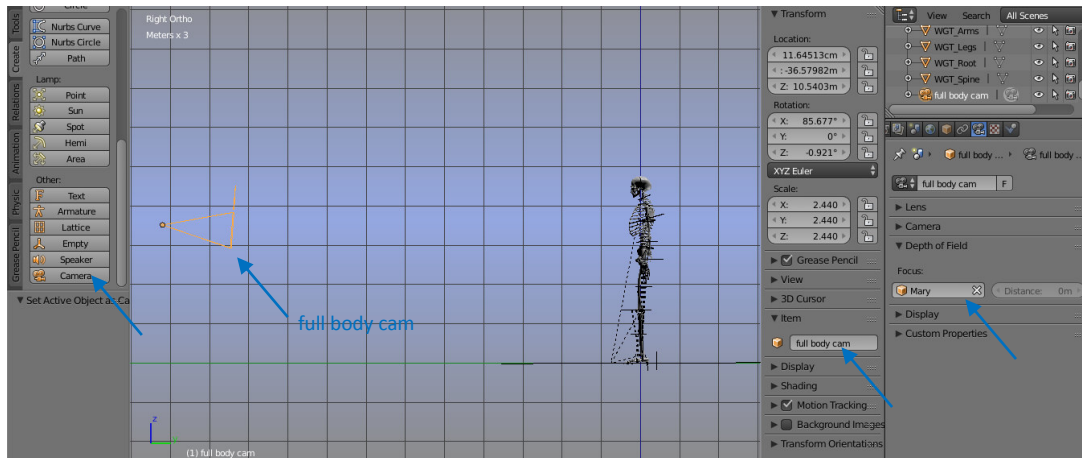


Abb. 52: Positionierung der full body cam

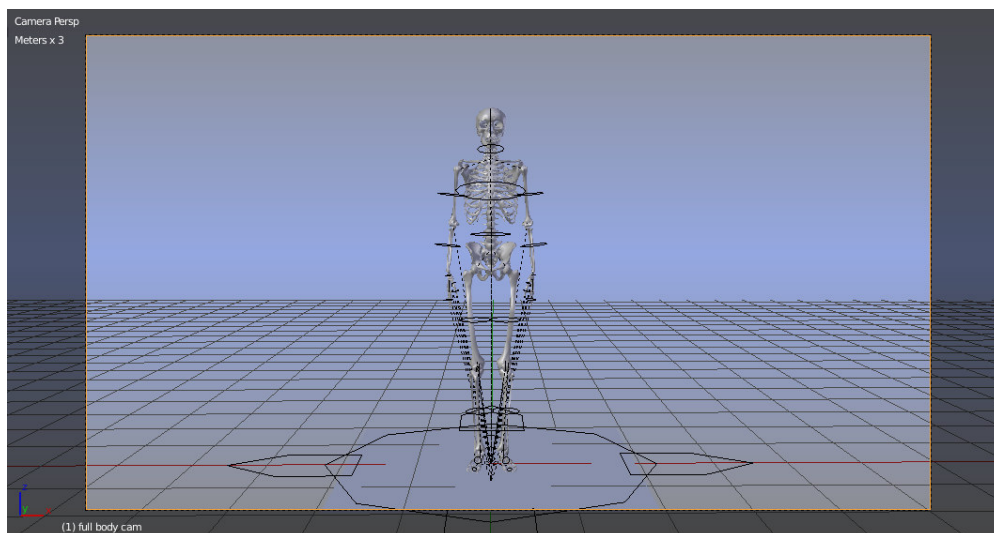


Abb. 53: Aus der Kameraperspektive: full body cam

Eine zweite Kamera wird für den Nahbereich, wie oben beschrieben, erschaffen (Abb. 54; 55). Dementsprechend wird die Kamera seitlich am Referenzgelenk positioniert. Die Schärfentiefe richtet sich am Hüftgelenk¹⁵ an Pelvis. Werden mehrere Kameras in Blender verwendet, so wird mit der Tastenkombination **[Strg] + [Pad0]** die zuvor selektierte Kamera aktiviert. Als Anhaltspunkt hierfür dient das gefüllte Dach

¹⁵ Die Kameraeinstellung am Kniegelenk ist im Kapitel 20 nachzulesen.

über der aktiven Kamera. Damit die Detailkamera auf das Hüftgelenk während einer ausführenden Bewegung fixiert ist, wird eine Parent-Child-Verbindung zwischen der Kamera und dem CTRL-Hip-Bone erschaffen (nacheinander mit **[Alt]** + **[K]** CTRL-Hip-Bone und Kamera auswählen, **[Strg]** + **[P]** und Bone). Blender positioniert die Kamera am Boden. Diese muss auf Höhe des Hüftgelenks verschoben werden. Die Detailkamera führt nun eine Parallelfahrt zu Pelvis durch und zeigt die Bewegung des Gelenks an. Im Item wird die Kamera in beispielsweise zoom cam umbenannt.

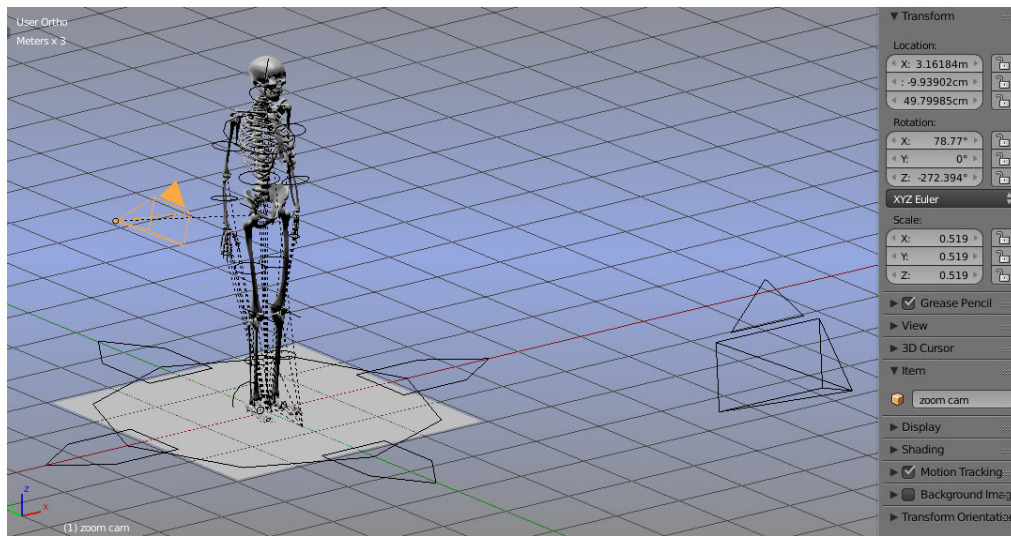


Abb. 54: Positionierung der zoom cam

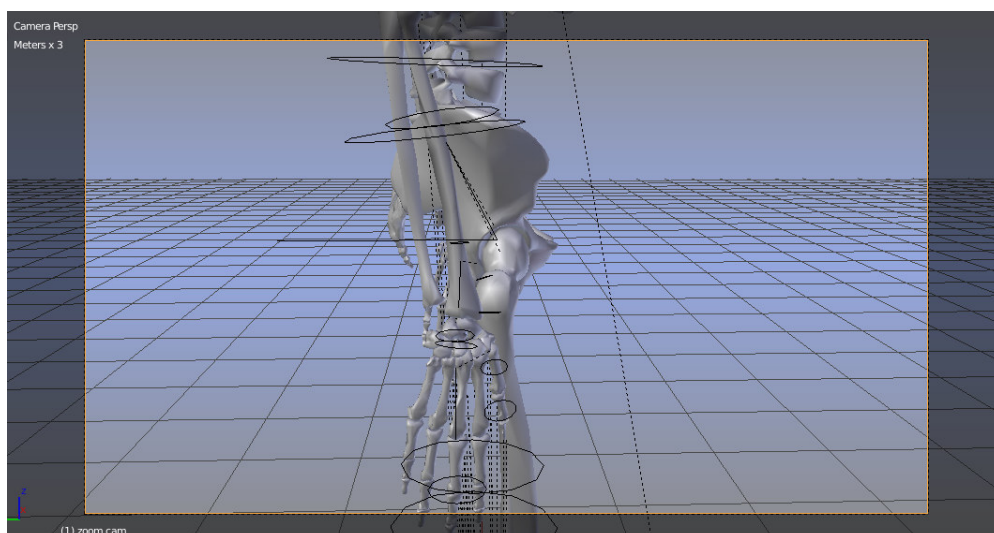


Abb. 55: Aus der Kameraperspektive: zoom cam

15.2 Licht

Im Tool shelf wird unter Create die Lichtquelle Sun gewählt. Diese zeichnet sich durch ihre parallele, starke Ausstrahlung aus (Beck 2015, S. 312). Es werden zwei Lichtquellen erzeugt und in sehr großer Distanz, zum Mesh positioniert. Dabei sind die Beleuchtungskörper (vor und hinter dem Skelett) schräg-erhöht ausgerichtet. Eine dritte Lichtquelle wird erzeugt und hinter die full body cam platziert (Abb. 56). Die Positionierungen der Sonnen gewährleisten ideale Ausleuchtung des Objekts. Im Properties Editor im Button Lamp wird die Energie der ersten beiden Sonnen auf 40%, der dritten Sonne auf 50% eingestellt. Um einen Schatten zu vermeiden, wird Shadow deaktiviert.

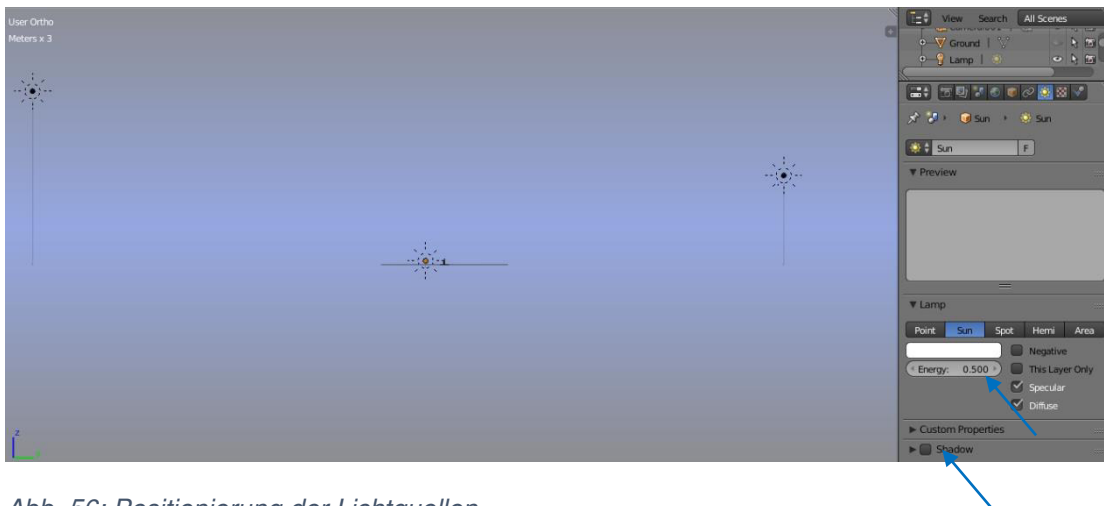


Abb. 56: Positionierung der Lichtquellen
Sagittalansicht

16 Erstellung der Kraftanzeige

Die Kraftsäulen werden in einer neuen Szene konstruiert. Im Szenen-Fenster mit + eine neue Szene erstellen, Copy Settings auswählen (Abb. 57) und die neu erschaffene Szene in beispielsweise overlay umbenennen.

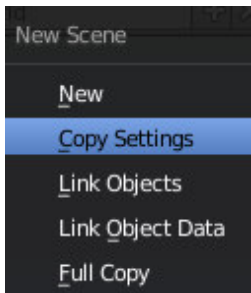


Abb. 57: New Scene

Die Kraftsäulen werden im x-y-Koordinatensystem erschaffen (**Pad 7**). Für die Ansicht auf die Diagramme, wird eine Kamera, wie im Kapitel 15.1 beschrieben, im Ursprungspunkt erstellt und auf der z-Achse nach oben verschoben. Gegebenenfalls mit dem Shortcut **Alt** + **R** die Kameransicht drehen sowie mit **S** größer skalieren (Abb. 58). Die Kameraansicht kann innerhalb der Kameraperspektive (**Pad 0**) nachjustiert werden.

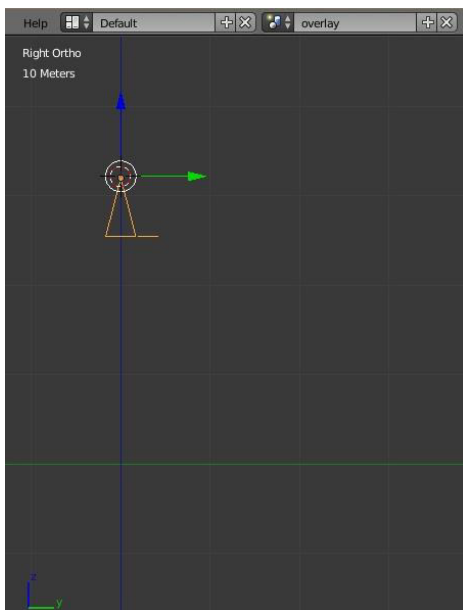


Abb. 58: Kamerapositionierung: Kraftsäulen

Die Säulen für die Kraftanzeige können in der Kameraperspektive erzeugt und bearbeitet werden. Dazu wird als erstes innerhalb des Object Modes im Ursprungspunkt eine neue Plane erzeugt (**Ctrl** + **S**, Cursor to select, **Ctrl** + **A**). Diese Plane wird mit der Kantenfunktion (in Edit Mode anwählbar) auf die gewünschte Größe skaliert und an den Rand der Kameransicht verschoben. Diese Fläche wird selektiert, dupliziert

und zum gegenüberliegenden Rand verschoben ($\square + D + X$). Die beiden Flächen können in Item in beispielsweise *underground plane l./r.* umbenannt werden. Die Färbung wird im Properties Editor im Icon Material und Add a new material festgelegt. Unter Diffuse ist es möglich, die Farbe für die Untergrundsäulen auszuwählen, hier anthrazit und unter Shading und Shadeless einen Schattenverlauf deaktivieren. (Abb. 59).

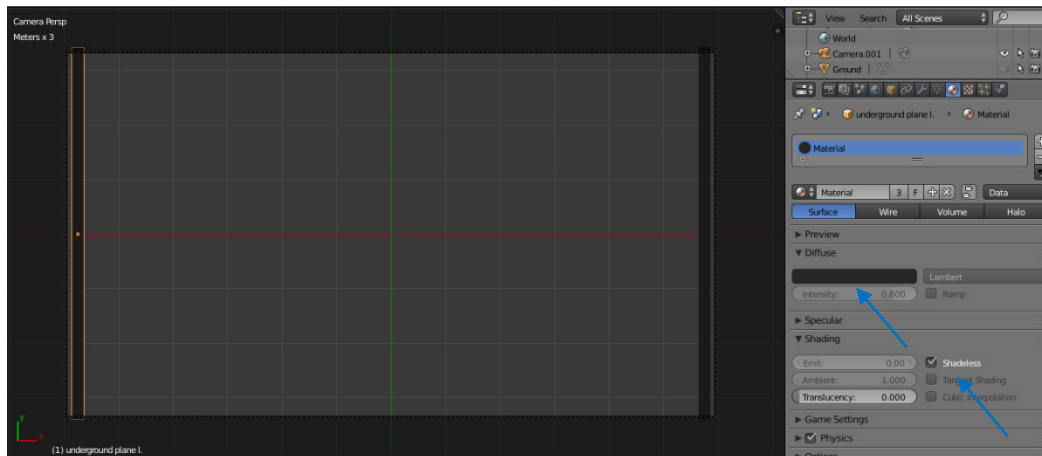


Abb. 59: Kraftsäulen: *underground plane*

Es werden, wie oben beschrieben, eine weitere Fläche pro Seite konstruiert. Diese beiden Flächen werden jeweils in 2cm Höhe über der *underground plane* positioniert (Properties, unter Transformation die Location in x-Richtung auf 2cm bestimmen). Die Länge dieser neuen Balken muss nicht identisch mit den darunterliegenden Flächen sein. Das liegt daran, dass die neu erstellten Flächen als Anzeige der einwirkenden Kraft am Gelenk dienen und ihre Länge später in der Animation dementsprechend ausgerichtet werden. Die Farbe wird auf Rot bestimmt. Benannt werden diese in beispielsweise *upper plane r./l.*" (Abb. 60).

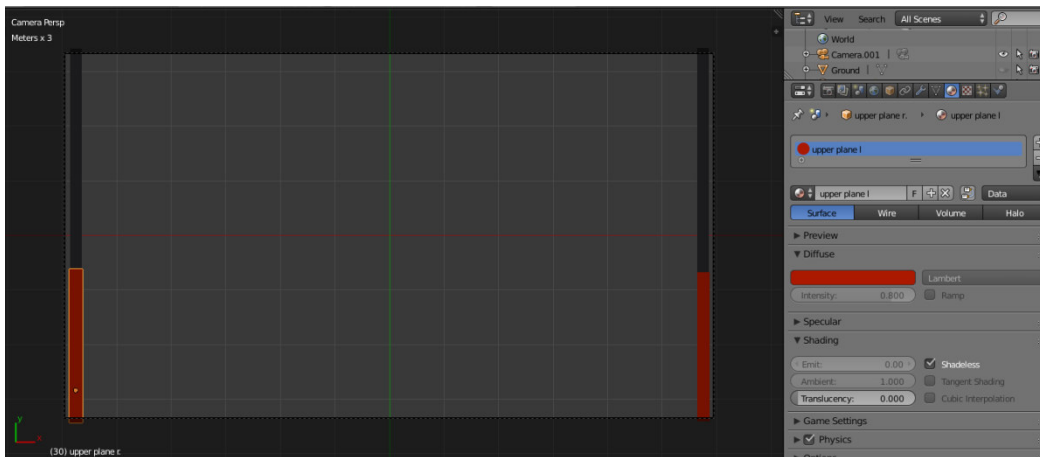


Abb. 60: Farbgebung der Kraftsäulen: upper plane

Im Anschluss darauf, werden im Objekt Mode die Beschriftungen konstruiert. Der Cursor wird am Ursprungsort gesetzt. Im Tool shelf wird unter Create Text ausgewählt. Im Edit Mode wird die Überschrift dieser overlay scene eingefügt, hier Force display of the hip. Im Object Mode wird der erstellte Text ausgerichtet. Im Properties Editor unter Data Object im Fenster F das Textfeld benennen, zum Beispiel heading. Unter Font wird die Größe geändert. Den Text im ausgewählten Bereich positionieren und im Fester ausrichten (Abb. 61). Die Beschriftung für die Kraftsäulen werden ebenso erstellt (linke Kraftanzeige: Left hip F [% BW], rechte Kraftanzeige: Right hip F [% BW] Reference limb, Abb. 61).

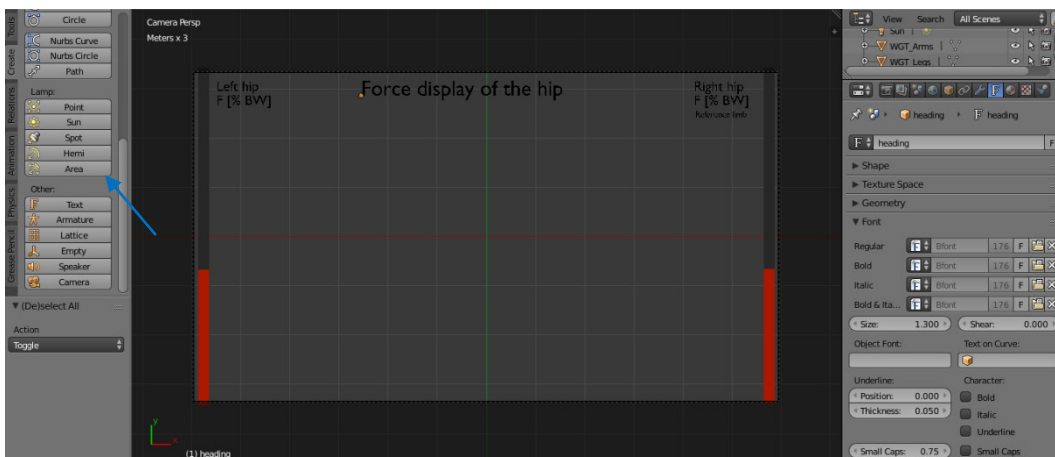


Abb. 61: Kraftbalken: Textpositionierung

Die Erstellung einer Skala der Säulendiagramme erfolgt ebenfalls mit dem Text-Objekt. Die Unterteilungslinien werden in einem Abstand von 2,5cm über der underground plane gesetzt. Die Maßeinteilung erfolgt in 50er Schritten, wobei nur die 100er ausgeschrieben werden (Abb. 62).

Bei der späteren Programmanwendung soll während des Einbeinstands, nur die Anzeige für das Referenzbein zu sehen sein. Daher wird für die Animation der Kraftanzeige im Einbeinstand eine weitere Plane benötigt. Diese wird in Form einer breiten Fläche links neben der Kameraperspektive in einer Höhe von 3cm platziert. Die neu erzeugte overlay plane ist in ihrer Länger größer als die angezeigte Kameraperspektive Die Farbe entspricht dem Hintergrund der world-Szene (Kap. 13, Abb. 62).

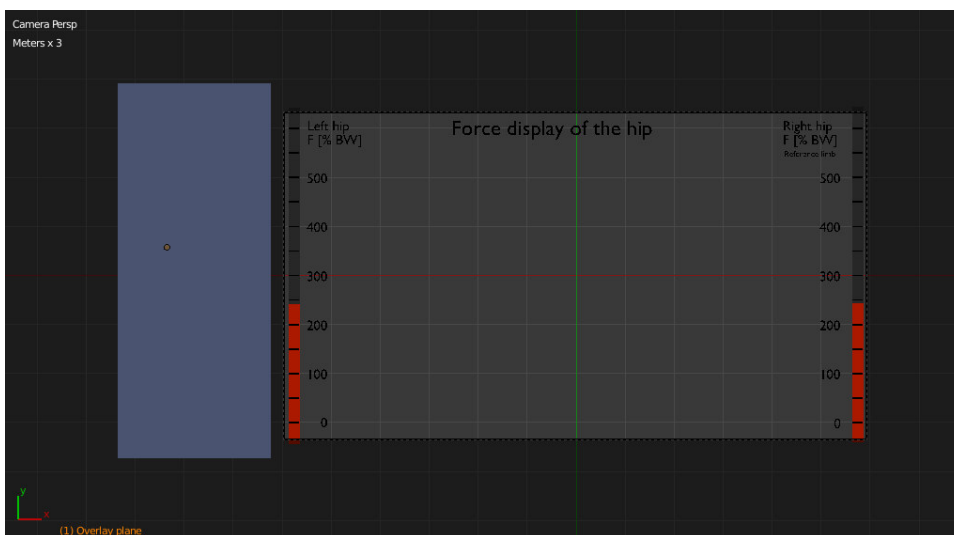


Abb. 62: Kraftsäulen inklusive Beschriftung und overlay plane

17 Break

Für eine Pausenfunktion wird eine weitere Szene erschaffen und in break umbenannt. Eine Kamera wird, wie im Kapitel 15.1 beschrieben, erzeugt und positioniert. Die Beschriftung break wird für den späteren Erhalt der Ansicht auf das Skelett unten rechts am Rand platziert (Abb. 63).

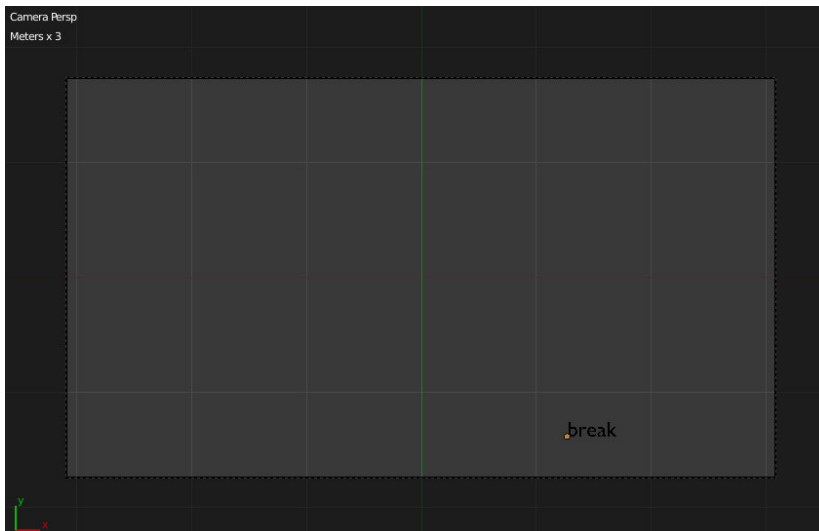


Abb. 63: Break, Kameraperspektive

18 Animation der Kraftanzeige vom Hüftgelenk

Zunächst soll die Kraftanzeige am Hüftgelenk konstruiert werden. Es wird demnach am hip-Programm gearbeitet. Die Animation der Kraftanzeige für das Kniegelenk erfolgt gleich der Animation am Hüftgelenk mit den entsprechenden Werten aus Kapitel 21.

Im Screen Layout wird von Default in Animation gewechselt. Für jede animierte Aktion in der world-Szene werden in der overlay-Szene jeweils die rechte und die linke Säule einzeln animiert. Eine Ausnahme bildet der Einbeinstand. In dieser Aktion werden die rechte Kraftanzeige sowie die overlay plane ausgerichtet. Insgesamt werden 10 neue Aktionen, wie im Kapitel 12 beschrieben, erstellt: scale stand r., scale stand l., scale walk r., scale walk l., scale slow motion r., scale slow motion l., scale to crouch down r., scale to crouch down l., scale one leg stand r., overlay plane.

Die Keyframes sollen diesmal nicht automatisch von der Software gespeichert werden, sondern manuell über den Befehl `I` + LocRotScale.

Die Ausrichtung der Kraftanzeigen erfolgt entsprechend der Fachliteratur (Bergmann 2004, S. 24, ders. et al. 2001, S. 867).

Die Zeitspanne der einzelnen Aktionen der overlay-Szene wird an die Zeitspanne der zugehörigen Aktionen der world-Szene angepasst. Auch die Keyframe-Positionierung an der rechten Kraftsäule wird entsprechend der Keyframes der world-Szene gesetzt.

Für die Standanimation des Hüftgelenks wird in einer neuen Aktion scale r. zunächst die rechte Kraftanzeige animiert. Die Zeitspanne beträgt insgesamt 41 Frames. Bei Frame 1 wird im Edit Mode die untere Säulenkante der upper plane mit der Kantenfunktion selektiert und der Cursor an der ausgewählten Kante platziert (\uparrow + S , Cursor to select). Im Object Mode wird die upper plane an das gewünschte Maß skaliert (S + Y), hier 75 % BW (Bergmann 2004, S. 23). Anschließend sind die Änderungen über I + LocRotScale der Keyframes zu speichern. Da bei der Standanimation keine Kraftänderung stattfinden soll, wird Frame 1 kopiert und an Frame 41 eingefügt. Die linke Kraftsäule wird nun selektiert und entsprechend dem rechten Balken animiert (Abb. 64).

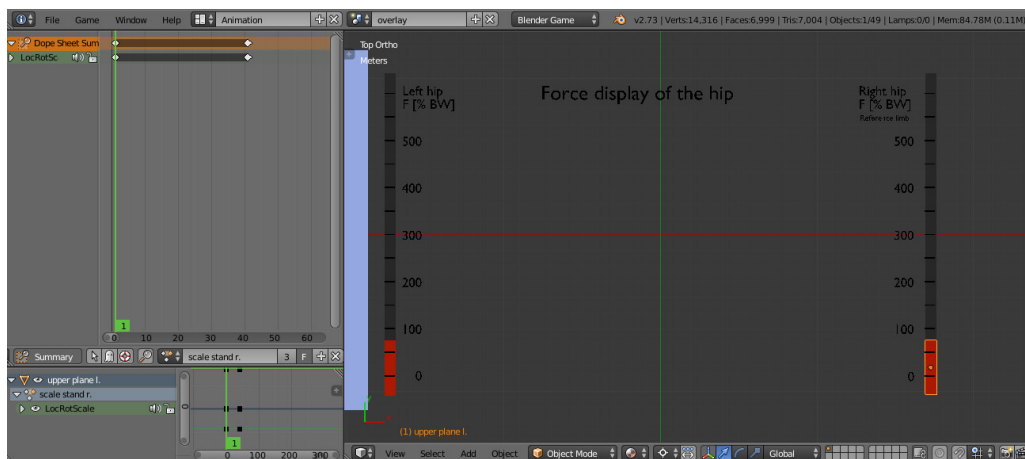


Abb. 64: Animation: Kraftanzeige vom Hüftgelenk, stand

Eine neue Animation der Kraftsäulendiagramme wird für die Kraftanzeige am Hüftgelenk während des Gehens benötigt. Die Zeitspanne der Frames beträgt hier, ebenso wie in der Animation des Gangzyklus, 39 Frames. Die Ausrichtung der rechten Kraftsäule erfolgt entsprechend der entnommenen Werte aus der Fachliteratur (Bergmann 2004, S. 24). Die Keyframes werden an den entsprechenden Frames der Gangphasen aus Kapitel 12 gesetzt (Tab. 4; 5). Frame 1 wird, wie in der world-Szene, ebenfalls an Frame 41 kopiert (Abb. 65).

In der world-Szene wird das kontralaterale Bein in seinen Gangphasen betrachtet. Die Keyframepositionierung der linken Kraftanzeige ergibt sich aus dieser Betrachtung (Tab. 5, Abb. 66). Um einen flüssigem Übergang in der Bewegungsschleife zu erlangen, wird an Frame 1 sowie Frame 41 ein Keyframe mit einer ausgerichteten Kraft von ca. 140% BW platziert.

Die Kraftsäulen für die slow motion-Animation werden entsprechend der Kraftbalken von walk animiert. Die Framepositionierungen sind in der Tabelle 5 dargestellt.

Gangphase	Frame Referenzbein	Krafteinwirkung [% BW]	Frame kontralaterales Bein
Initial contact		75	
walk	1		21
slow motion	1		61
Loading response		220	
walk	4		25
slow motion	10		72
Mid stance			
F_{max}		238	
walk	7	-	28
slow motion	19	210	81
mid Mid stance¹⁶	10		32
späte Mid stance			
walk	16		35
slow motion	46		102

¹⁶ In der mid Mid stance interpoliert Blender den Wert der Kraftanzeige.

Terminal stance		190	
walk	21		37
slow motion	61		108
Pre-swing		113	
walk	25		4
slow motion	72		10
Initial swing		55	
walk	31		6
slow motion	85		16
Mid swing		25	
walk	35		9
slow motion	102		25
Terminal swing¹⁷		75	
walk	38		-
slow motion	111		-

Tabelle 5: Ausrichtung der Kraftbalken am Hüftgelenk, walk, slow motion

¹⁷ Auf der kontralateralen Seiten liegen die Frames von Terminal swing sowie Initial contact sehr nahe bei einander. An dieser Stelle wird die angezeigte Kraft mittels der Software interpoliert.

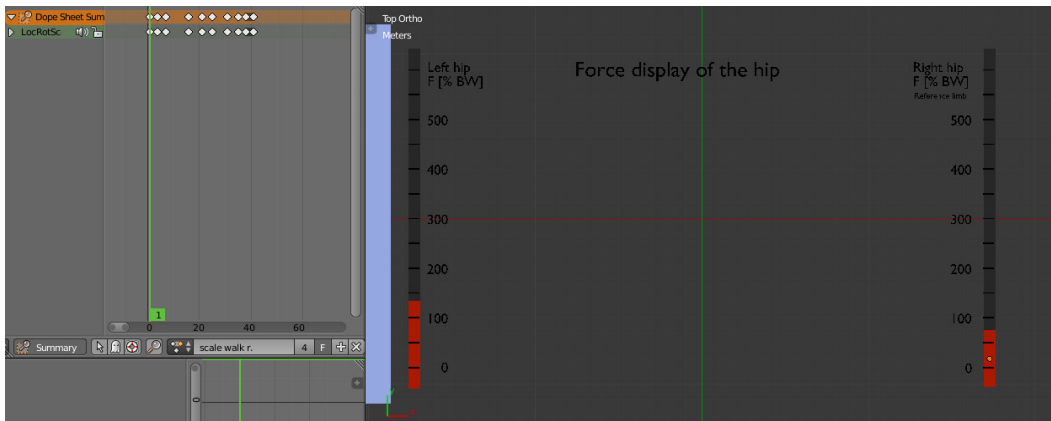


Abb. 65: Animation: Kraftanzeige vom Hüftgelenk, Frame 1, walk

Die Kraftanzeige für das einbeinige Stehen wird am Referenzbein ausgerichtet, somit an der rechten Kraftsäule. Eine angezeigte Kraft von 231% BW wird konstruiert (Bergmann et al. 2001, S. 867).

Damit das linke Diagramm bei der Programmausführung des Einbeinstands nicht sichtbar ist, wird dieses Diagramm von der overlay plane verdeckt. Die Positionierung dieser Plane ist von Frame 1 bis beispielsweise Frame 135 unverändert, so dass die Plane außerhalb der Kameraperspektive bleibt. Die Plane muss an dieser Stelle während der ersten Frames außerhalb der Kameraperspektive bleiben. Bei Frame 140 wird overlay plane so platziert, dass diese die linke Kraftsäule mitsamt der Beschriftung verdeckt. Diese Position wird abgespeichert, an Frame 230 kopiert und eingefügt (Abb. 66). Die Zeitspanne entspricht der Animationszeit vom one leg stand.

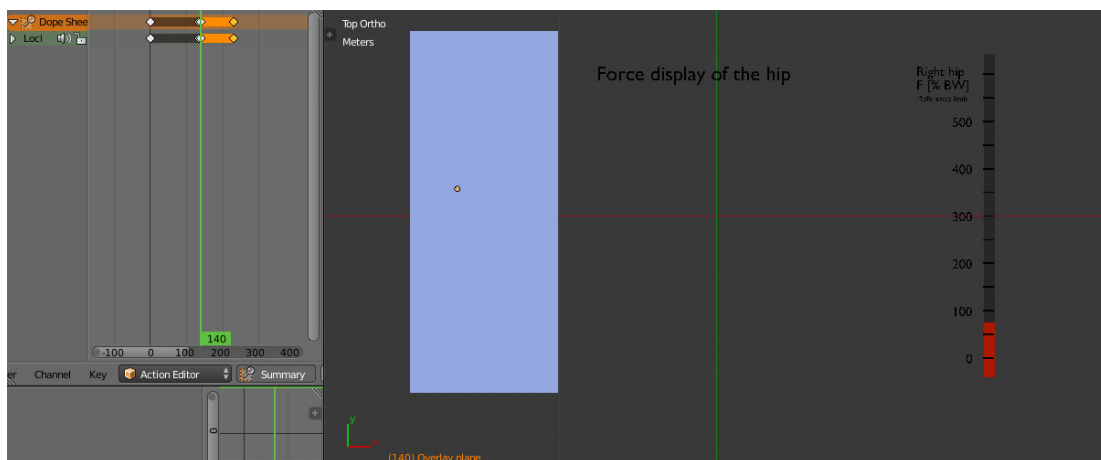


Abb. 66: Animation: Kraftanzeige vom Hüftgelenk, Frame 140, one leg stand

19 Start-Datei

In der start-Datei werden die Menü- sowie die Manual-Ansicht erstellt. Das Programm „Motion and force display“ startet mit der Menü-Ansicht. Hierüber lassen sich die Unterprogramme anwählen und in die Manual-Ansicht wechseln. In der Manual-Ansicht ist die Belegungsanzeige der Tasten dargestellt.

Die Menü- und Manual-Ansicht wurden mit einem externen Programm angefertigt und werden in Blender eingefügt (s. weiter unten, Abb. 67; 69). Die Anwahl-Buttons sind ebenfalls einzeln mit dem externen Programm erstellt worden (Abb. 68; 70).



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Created by:
Gerwin Reimers
Weronika A. Krzywik-Groß

Abb. 67: Menü-Ansicht

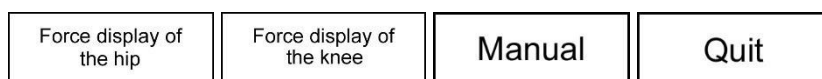


Abb. 68: Menü-Ansicht: Buttons

Number pad

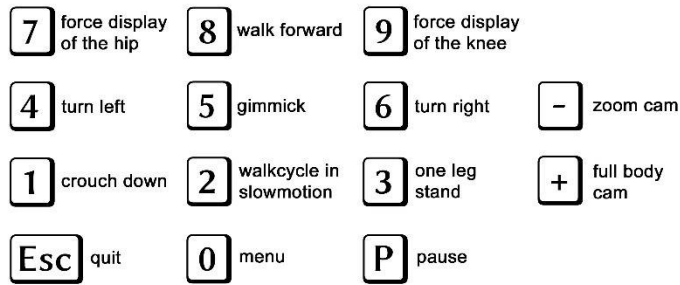


Abb. 69: Manual-Ansicht



Abb. 70: Manual-Ansicht: Button

Eine neue Blender-Datei wird geöffnet und die Szene in menu umbenannt. Das enthaltende Objekt wird entfernt. Die Kamera wird in erhöhter Position über den Ursprungspunkt platziert. Es werden insgesamt drei Lichtquellen in Form von Sonnen verwendet. Die erste Lichtquelle ist erhöht und befindet sich seitlich von der Kamera mit einer Energie von 65%. Die zweite Lichtquelle befindet sich über der Kamera mit einer Energie von 10% und die dritte Lichtquelle auf der entgegengesetzten Seite der Kamera besitzt eine Energieeinstellung von 25% (Abb.71).

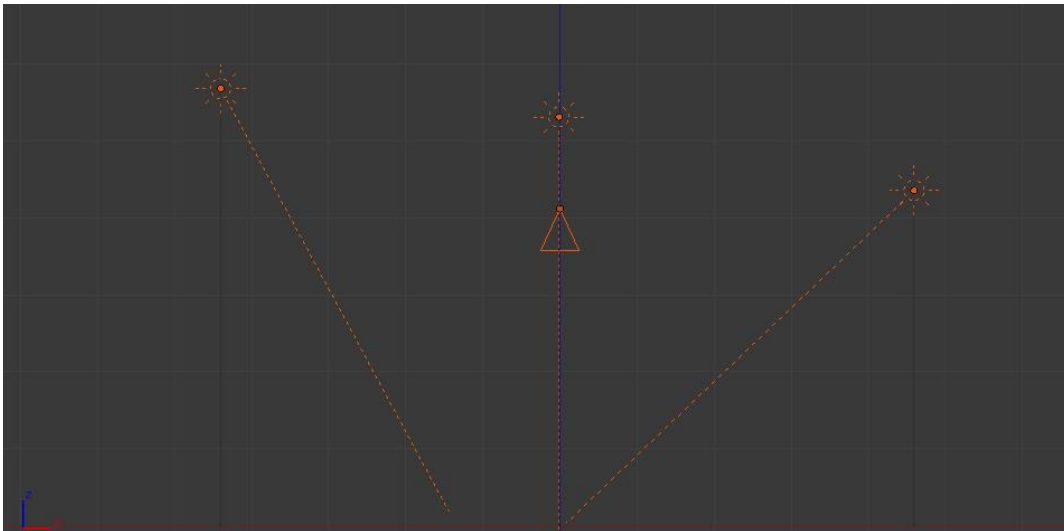


Abb. 71: start-Datei: Positionierung der Kamera sowie der Lichtquellen

Um ein Bild als jpg-Dateiformat in Blender zu importieren, wird in den User Preferences die Einstellung Import-Export: Import Images as Planes aktiviert. Diese Einstellung ist unter File, User Preferences bei den Add-ons (runter scrollen) zu finden (Abb. 72; 73; 74).

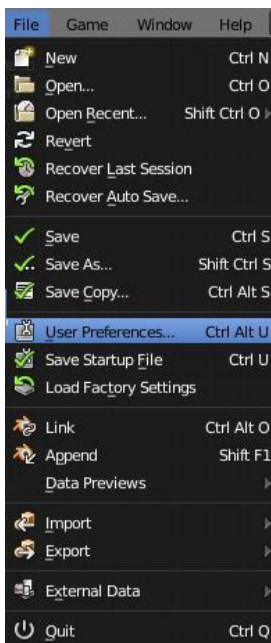


Abb. 72: File – Fenster: User Preferences

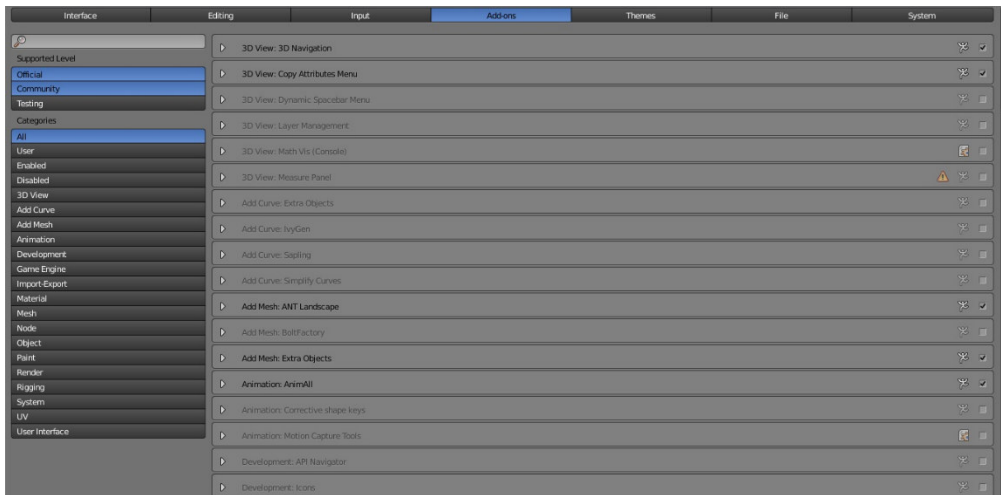


Abb. 73: User Preferences – Fenster: Add-ons

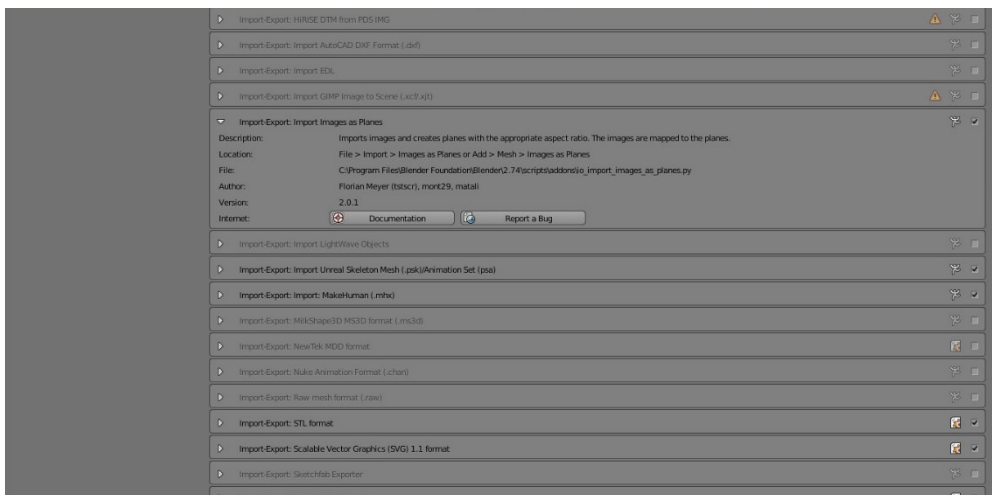


Abb. 74: Add-ons – Fenster: Import-Export: Import Images as Planes

Jetzt kann die Menü-Ansicht sowie die zugehörigen Buttons über File, Import und Images as Planes eingefügt werden (Abb.75).

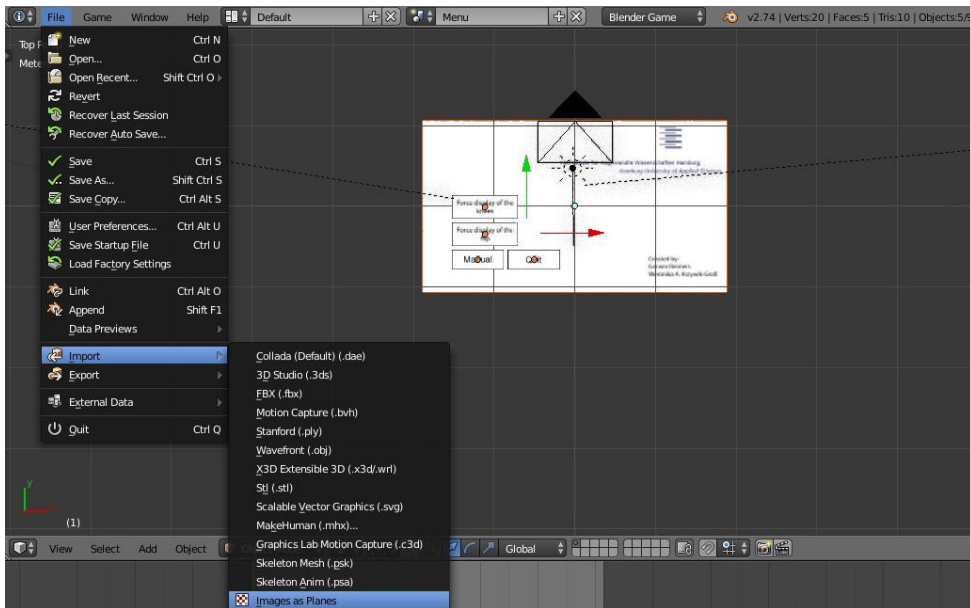


Abb. 75: Import: Menü-Ansicht

Die externen Dateien werden automatisch am Ursprungspunkt platziert. Diese werden nun ausgerichtet (Abb. 76). Optional kann in die Kameraperspektive gewechselt werden (Pad0). Unter Item wird diese Bildunterlage beispielsweise in background umbenannt. Im Properties Editor wird der Kamerafokus auf background eingestellt. Wie bekannt wird für die Menü-Ansicht samt Buttons die Funktionseinstellung Shadeless aktiviert.

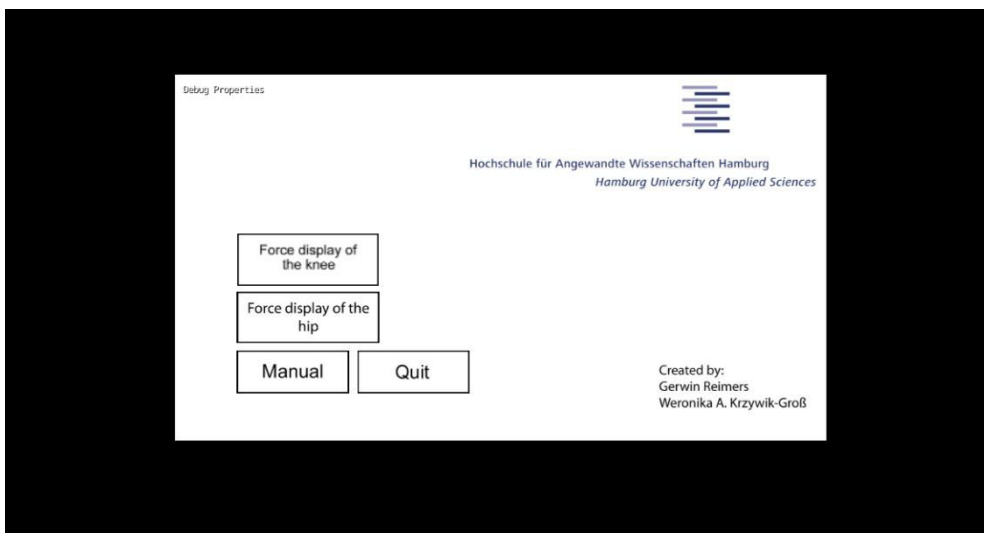


Abb. 76: menu-Ansicht

Die Manual-Ansicht wird in einer neuen Szene (manual), wie oben beschrieben, erstellt (Abb. 77).

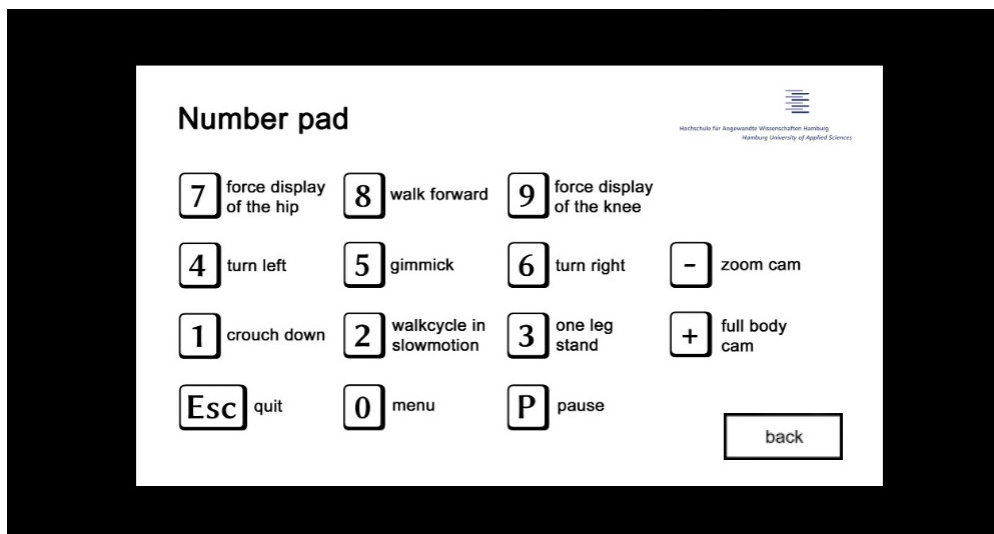


Abb. 77: manual-Ansicht

In der Visualisierung soll in den beiden Ansichten mittels der Computermaus auf die Buttons zugegriffen werden können. Dafür wird jeweils in beiden Szenen (menu und manual) ein Skript im Text Editor erstellt. Dieser ist in der Game Logic über dem Properties Editor zu finden. Im Window Header wird mit dem Button +New ein neues Textdokument erstellt und in mouse on umbenannt (Abb. 78; 79).



Abb. 78: Text Editor: Window Header

Im Textfeld wird der Befehl für die Anzeige des Mauszeigers eingegeben (Abb. 79):
import Rasterizer
Rasterizer.showMouse(1)

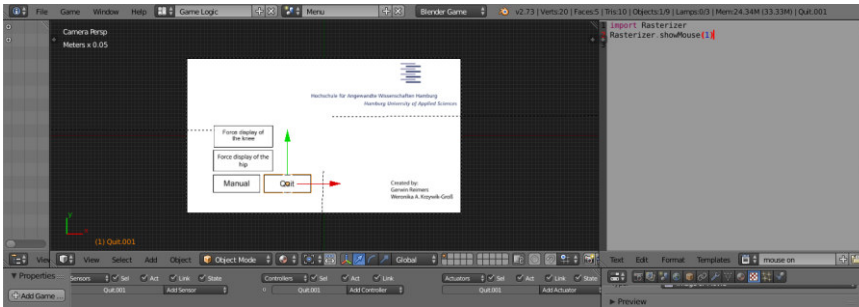


Abb. 79: mouse on: Mausanzeige aktiv

Ein neues Textdokument mit der Bezeichnung mouse off wird mit dem +-Button erzeugt und der Befehl für die Inaktivierung der Mausanzeige eingegeben:

```
import Rasterizer
Rasterizer.showMouse(0)
```

Dieselben Schritte werden nun für die zweite Szene ausgeführt.

20 Game Engine

Mittels der Game Engine wird eine interaktive Funktionssteuerung erstellt. Begonnen wird in dem hip-Programm. Im Screen Layout wird vom Default in Game Logic gewechselt, die Visualisierung der Bewegungsabläufe erfolgt in der world-Szene im Pose Mode. Im Logic Editor werden Objekte mit Animationen, Interaktionen sowie Verhaltensweisen versehen (Abb. 80) (Wartmann 2011, S. 335ff.) .

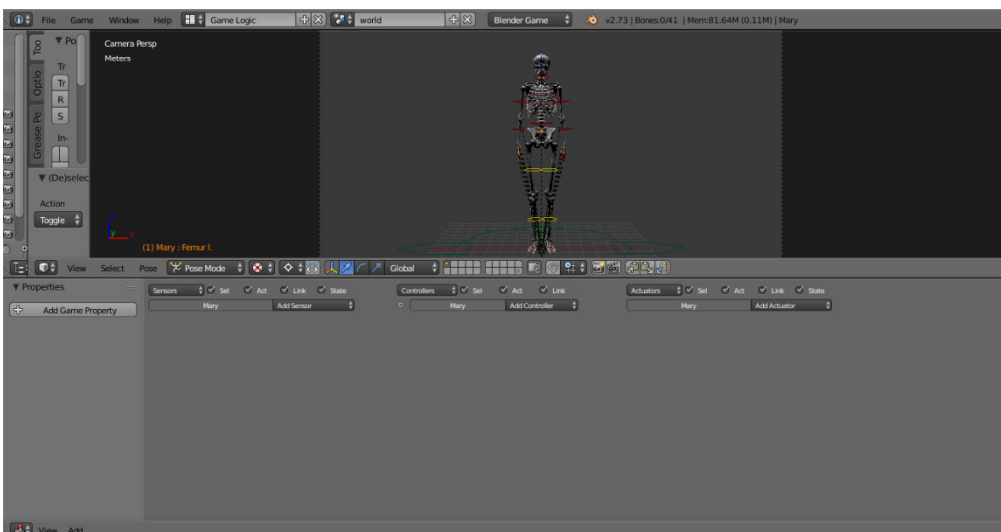


Abb. 80: Logic Editor

Damit die jeweiligen Animationen mithilfe der Game Engine abgerufen werden können, sind folgende Schritte notwendig:

1. Während der Anwendung des Programms „Motion and force display“ soll mit der Taste **Pad 0** in die Menü-Ansicht gewechselt werden:

Im Add sensor-Menü wird ein Keyboard-Sensor ausgewählt. Der Keyboard-Sensor wird mit menu betitelt. Im Fenster key wird der Tastaturcode gewählt, hier **Pad 0**. Der Keyboard-Sensor reagiert auf im Key-Fenster eingetragene Taste. Im Controller-Menü wird der Controller And ausgewählt, der einen Impuls weiter leiten, wenn alle angeschlossenen Sensoren einen positiven Impuls übermitteln. Der Ausgang des Keyboard-Sensors wird mittels gehaltener **⌘** mit dem Eingang des And-Controllers verbunden. Im Add Actuator-Menü wird der Auslöser gewählt, hier Game. Diese wird im Fenster daneben ebenso mit menu betitelt. Mit dem Game-Actuator werden Blender-Dateien nachgeladen, gestartet oder beendet. Jetzt wird im Fenster Game start game from the file gewählt und der Pfad der start-Datei ausgesucht (Abb. 81). Der Ausgang des Controller-Menüs wird mit dem Eingang des Add Actuator-Menüs verbunden.

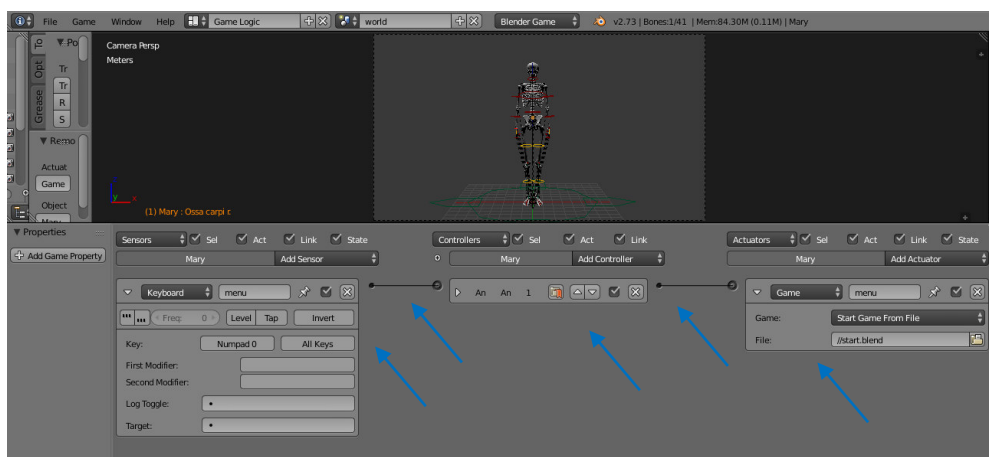


Abb. 81: Game Logic – world-Szene: menu

Somit erfolgt in der interaktiven Steuerung mittels **Pad 0** der Wechsel zur Menü-Ansicht.

2. Das Beenden des Programms „Motion and force display“ über die `[Esc]`-Taste

Für das Beenden des Programms „Motion and force display“ wird, wie oben beschrieben, verfahren. Betitelt wird dieser Sensor mit quit und als Tastaturcode wird die `[Esc]`-Taste bestimmt. Im Add Actuator-Menü wird ebenfalls Game ausgewählt und im Fenster Game Quit Game gewählt (Abb. 82).

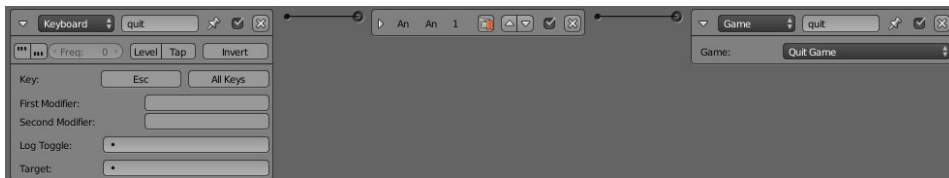


Abb. 82: Game Logic – world-Szene: quit

3. Die Rotationsbewegung des Skeletts bestimmen

Für die interaktive Steuerung des Skeletts werden zwei weitere Keyboard-Sensoren jeweils in Verbindung mit einem And-Sensor im Controller-Menü erzeugt. Die Linkssteuerung wird mit dem Tastencode `[Pad 4]`, die Rechtssteuerung mit `[Pad 6]` deklariert. Im Add Actuator-Menü wird hier Motion ausgewählt. Hierdurch werden Objekte bewegt. Im Motion Typ-Fenster wird Simple Motion ausgesucht. Location- sowie Rotation-Achsen kommen zum Vorschein. Nur die z-Rotation wird auf -2 Grad (Linkssteuerung) bzw. 2 Grad (Rechtssteuerung) eingestellt (Abb. 83; 84). Die Zahl 2 gibt dabei die Geschwindigkeit an, mit der die Bewegung ausgeführt wird.

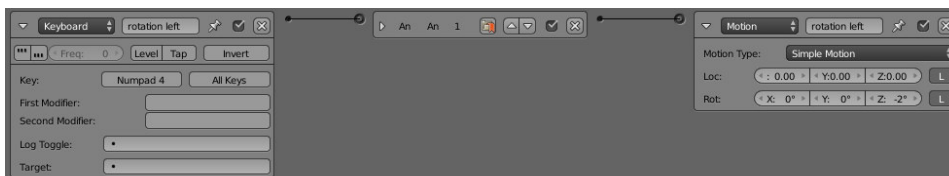


Abb. 83: Game Logic – world-Szene: rotation left

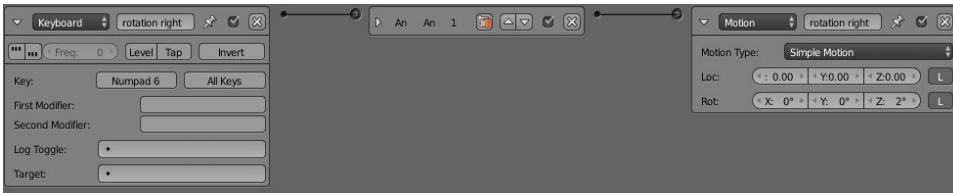


Abb. 84: Game Logic – world-Szene: rotation right

4. Wechsel zwischen der Knie- und der Hüftdarstellung

Während der Visualisierung soll es möglich sein zwischen der Knie- und hip-Darstellung wechseln zu können. Dieses erfolgt ähnlich der Menü-Realisierung, mit dem Unterschied in dem Tastaturcode. Mit der Taste **[Pad 9]** soll die Anzeige für die Kniegelenksbelastung ausgewählt werden und mit der Taste **[Pad 7]** die Belastung am Hüftgelenk. Dafür wird in der hip-Datei unter Add Actuator im Fenster File die Knie-Datei ausgesucht, in der Knie-Datei die hip-Datei. Näheres zur Knie-Datei siehe Kapitel 21.

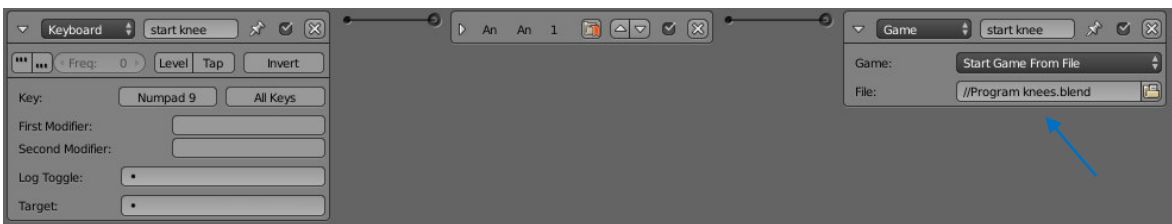


Abb. 85: Game Logic – world-Szene: Wechsel zum Knieprogramm

5. Visualisierung der Einbeinstandanimation

Nun wird die Visualisierung der Bewegungsabläufe realisiert. Am Beispiel des Einbeinstands wird dies ausführlich beschrieben:

Für den Einbeinstand wird im Keyboard-Sensor die Taste **[Pad 3]** bestimmt. Diesmal wird der Pulse-Mode aktiviert. Dieser befindet sich unter dem gewählten Sensortypen. Hier wird entschieden, welche Pulse, positive bei Aktivierung oder negative bei Deaktivierung, weitergeleitet werden. Der aktive Pulse-Mode (True-Zustand) bewirkt, dass so lange eine Taste gedrückt wird, hier die Taste **[Pad 3]**, positive Pulse rausgeschickt werden, woraufhin die Animation in einer Schleife abläuft. Andern-

falls würde der Bewegungsablauf nach einem kompletten Durchgang beendet werden. Die Frequenz (im Fenster neben dem Puls-Mode) bleibt auf null, da die auszuführende Bewegung direkt starten soll. Ein And-Sensor im Controller-Menü wird auch hier gewählt. Im Add Actuator-Menü wird diesmal ein Action-Typ ausgesucht. Dieser spielt die in der Animation angegebenen Aktionen ab, hier one leg stand. Die Zeitspanne, entsprechend der Aktion in der Animation: Frame 1 bis 90, wird angegeben. Der Action-Typ wird als Loop End deklariert. Das bedeutet, solange die eintreffenden Pulse positiv sind, wird die Schleife und somit der Bewegungsablauf weiter ausgeführt. Wird kein Puls übermittelt, wird die Schleife bis zum Ende ausgeführt und anschließend gestoppt. Auf der linken Seite unter Properties wird eine neue Eigenschaft mit dem + Add Game Property-Button und in one leg stand umbenannt. Der Datentyp Integer wird gewählt und auf 1 bestimmt. Damit wird festgelegt, dass die Darstellung der Bewegung sowie der Kraftanzeige (s. unten) synchron verlaufen und diese in einer Schleife wieder bei Frame 1 beginnen. Nun wird diese Eigenschaft dem Action-Typ im Fenster Frame Property mitgeteilt (Abb. 86).

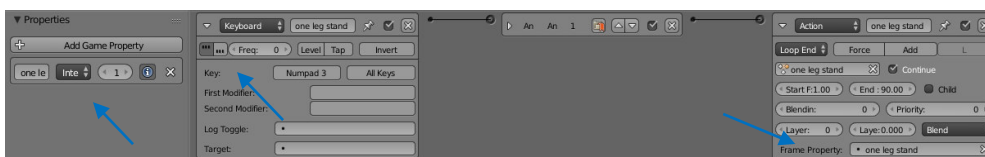


Abb. 86: Game Logic – world-Szene: one leg – Action-Typ

Damit ist der Bewegungsvorgang im Einbeinstand realisiert. Um neben dem Bewegungsablauf die Kraftanzeige zu visualisieren wird für die linke sowie für die rechte Anzeige jeweils ein Messenger aus dem Add Actuator-Menü erstellt. Mit diesem Actuator werden Nachrichten an andere Szenen (oder auch Objekte) gesendet. Eine Nachricht geht an die overlay plane in der overlay-Szene. So wird diese Message beispielsweise mit message op betitelt, im Objekt-Fenster wird die overlay plane ausgesucht und unter Subject die Nachricht mit einem Namen, beispielsweise overlay plane on, versehen. Die overlay plane soll in der Visualisierung im Einbeinstand die linke Kraftanzeige verdecken.

Die zweite Nachricht geht an die rechte Kraftsäule und wird mit message one leg stand betitelt. Im Objekt-Fenster wird die rechte Kraftsäule (upper plane r.) ausgewählt und im Subject die Nachricht in one leg stand benannt (Abb. 87).

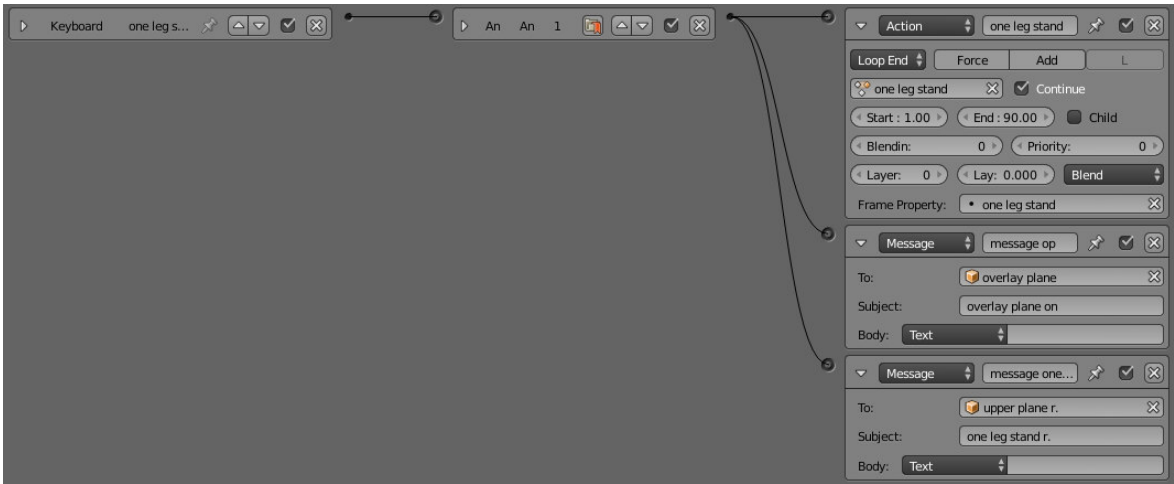


Abb. 87: Game Logic – world-Szene: one leg stand gesamt

Damit die Säulendiagramme synchron mit den Bewegungsanimationen ablaufen, werden Messenger verwendet

Die Realisierung der Bewegungsabläufe walk, slow motion, to crouch down wird, wie oben beschrieben und laut Abbildung 88, 89 und 90 erstellt, mit dem Unterschied jeweils in ein weiter zur Darstellung der linken Kraftanzeige.

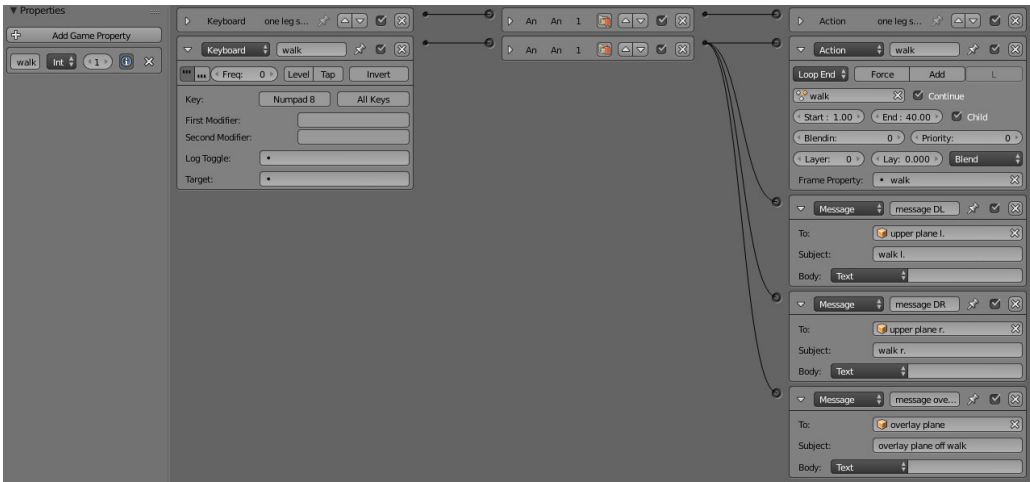


Abb. 88: Game Logic – world-Szene: walk

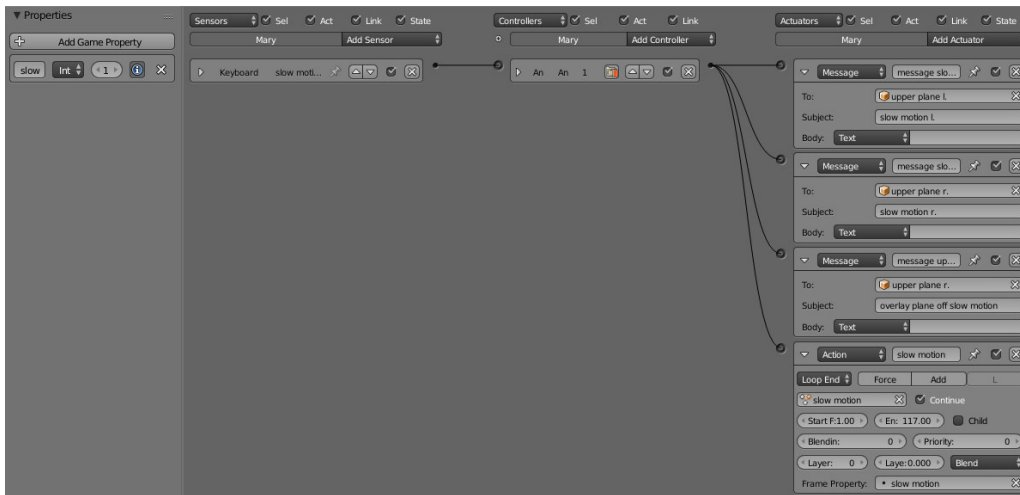


Abb. 89: Game Logic – world-Szene: slow motion

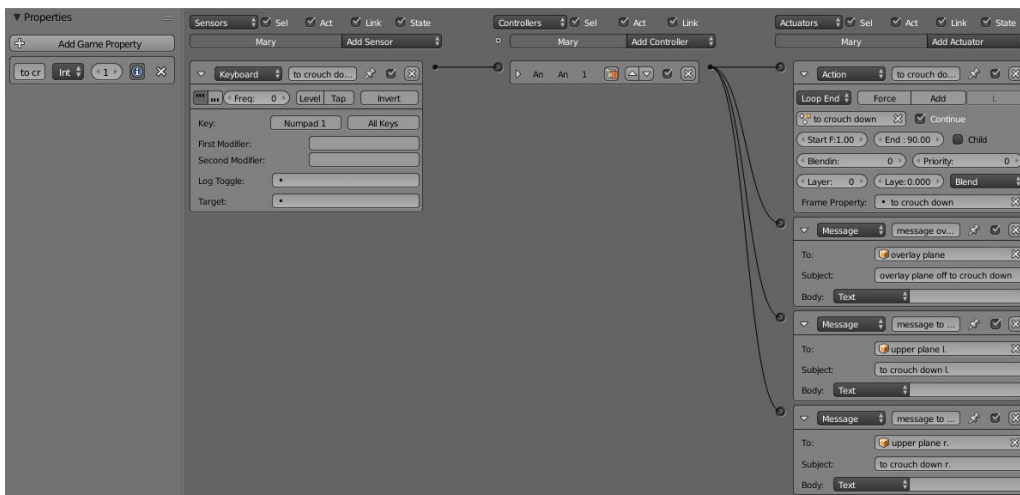


Abb. 90: Game Logic – world-Szene: to crouch down

Die Standposition und somit die Standanimation soll immer ausgeführt werden, solange keine Aktion durch Betätigen einer Taste stattfindet. Hierfür wird ein Always-Sensor gewählt. Mit diesem Sensortypen wird die Standanimation im Gegensatz zu den anderen Bewegungsanimationen ohne Tastenbetätigung ausgeführt (Abb. 91).

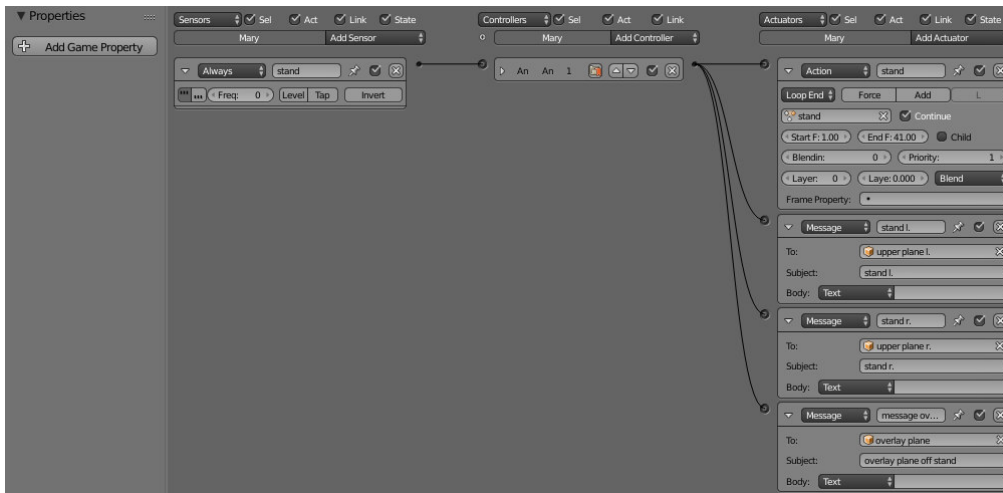


Abb. 91: Game Logic – world-Szene: stand

6. Realisierung des Kraftdiagramms zu entsprechenden Bewegungsanimationen

Die Kraftdiagramme sollen mit den Bewegungsabläufen synchron ablaufen. Dieses wird in der overlay-Szene verwirklicht. Hier wird die Darstellung der Kraftsäulen während eines Bewegungsablaufs realisiert. Dieses wird ausführlich anhand der rechten Kraftanzeige im Einbeinstand veranschaulicht. Die upper Plane wird dazu selektiert. Als Sensor wird hier Message ausgewählt, als Empfänger der Nachricht aus der world-Szene. Der Puls-Mode wird aktiviert, im Fenster Subject wird der Name der Sender-Nachricht aus der world-Szene eingegeben, hier one leg stand r.. Als Controller wird ebenfalls And ausgewählt. Es wird ein Action-Actuator des Typen Loop End bestimmt. Das Loop End bezieht sich auf den kompletten Durchlauf einer Bewegungsanimation. Die one leg stand-Action wird ausgewählt mit der entsprechenden Zeitspanne von 1 bis 90. Ebenso wie in der world-Szene wird unter Properties eine Eigenschaft des Datentyps Integer festgelegt, auf 1 deklariert sowie im Action-Actuator eingefügt (Abb. 92).

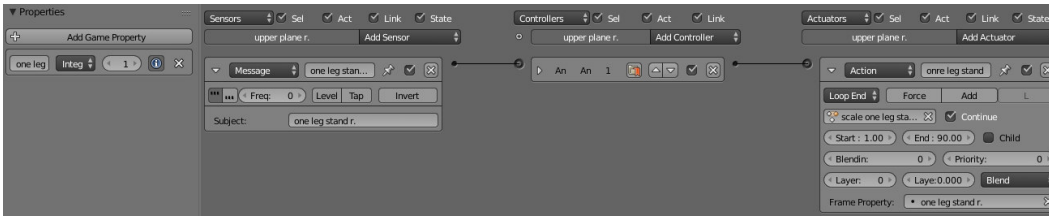


Abb. 92: Game Logic – overlay-Szene: rechte Kraftsäule, one leg stand

Das Visualisieren der Kraftanzeige in Abhängigkeit der Bewegungsabläufe to crouch down, stand, walk sowie slow motion wird entsprechend der oberen Beschreibung und der folgenden Abbildungen umgesetzt (Abb. 93).

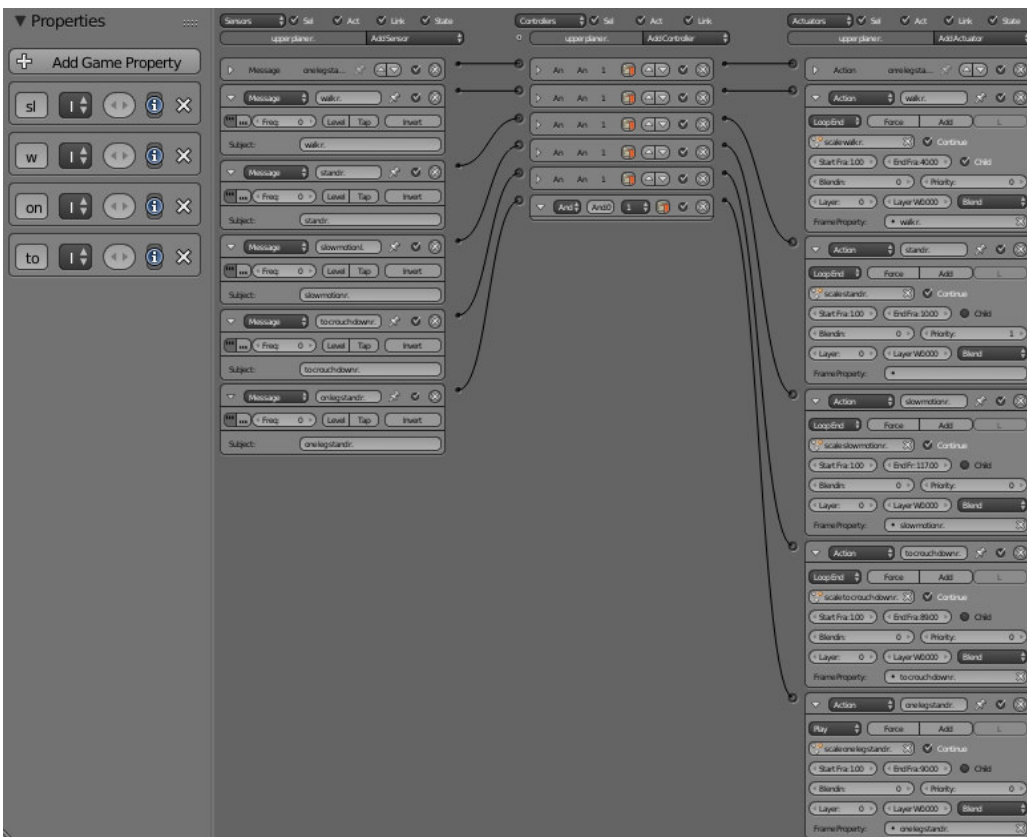


Abb. 93 Game Logic – overlay-Szene: rechte Kraftsäule, gesamt

Nun wird die linke Kraftanzeige selektiert und wie oben beschrieben für die einzelnen Bewegungsabläufe realisiert (Abb. 94). Hierbei wird kein Sensor für one leg stand benötigt. Aufgrund der Kraftdarstellung am Referenzbein während dieser Bewegung wird die linke Anzeige von overlay plane verdeckt.

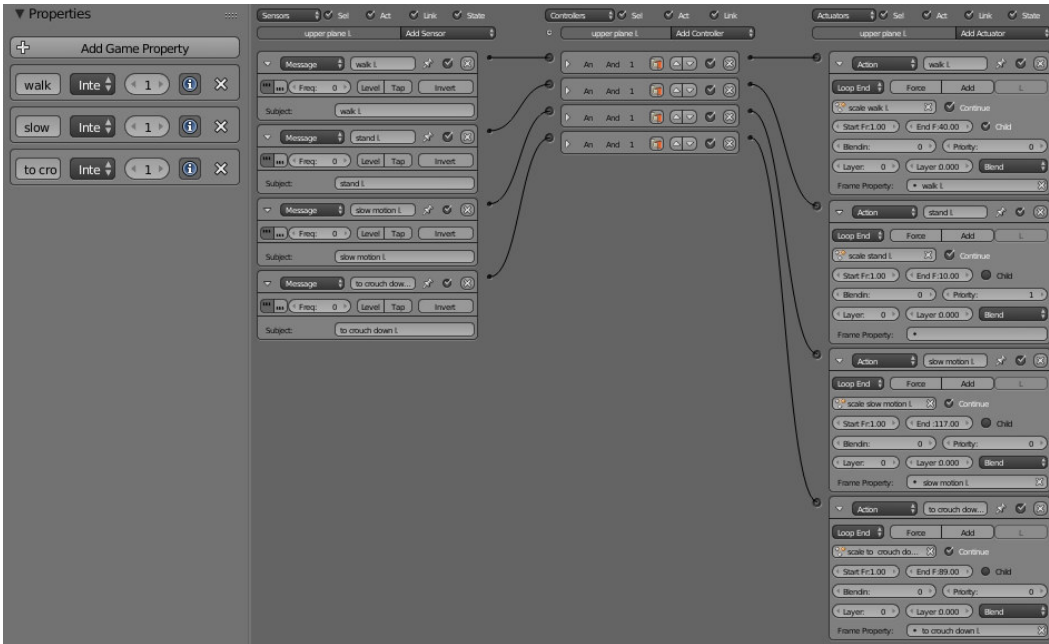


Abb. 94: Game Logic – overlay-Szene: linke Kraftsäule

Im Anschluss wird das Verhalten der overlay plane wie oben beschrieben realisiert. Diese soll im Einbeinstand die linke Säule verdecken, andernfalls unbewegt bleiben.

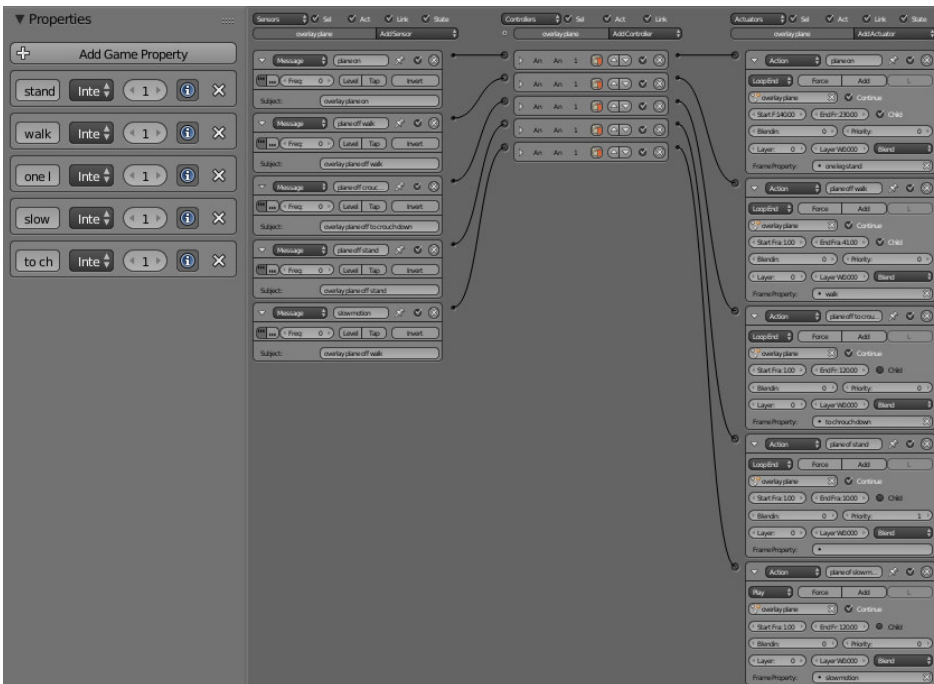


Abb. 95: Game Logic – overlay-Szene: overlay plane

7. Die Umsetzung der Pausenfunktion in der Game Engine

Die Umsetzung in der Game Logic für die Pausenfunktion erfolgt in der break-Szene. Sie wird wie folgt in Abbildung 96 realisiert.

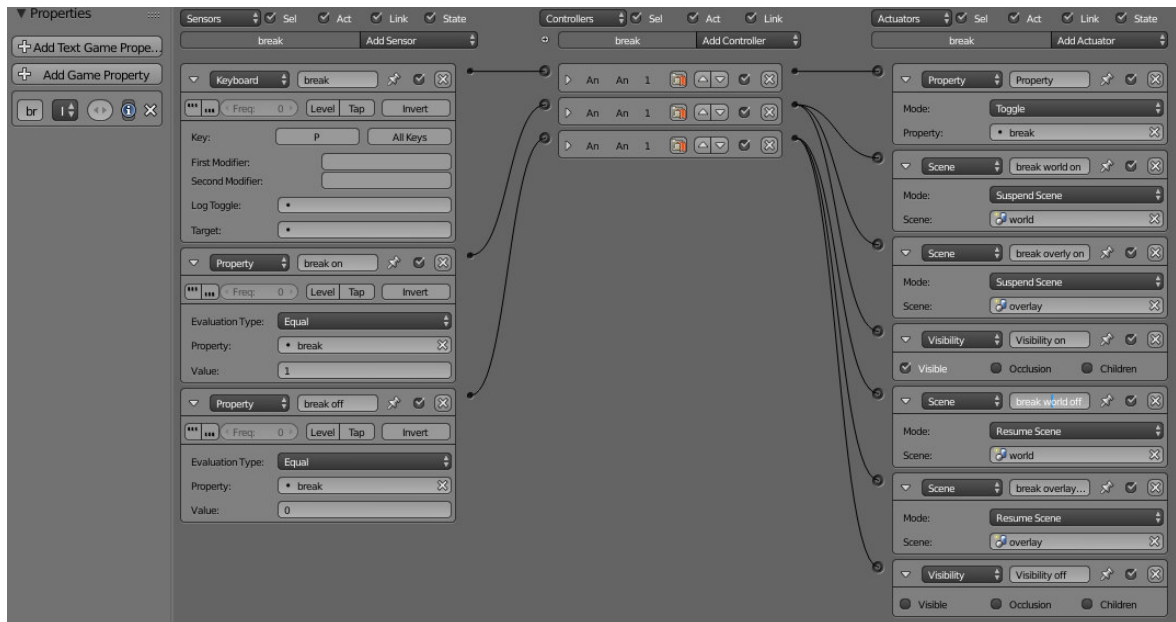


Abb. 96: Game Logic – break-Szene: break

Während der Visualisierung eines Bewegungsablaufs bei mehrmaliger Pausenbetätigung entsteht ein zeitlicher Versatz zwischen der angezeigten Kraft und dem Bewegungsverlauf. Es wird angenommen, dass die Konzeptionierung des Massengersystems dafür verantwortlich ist.

8. Die Game Logic in dem Startmenü

Um aus dem Startmenü heraus die einzelnen Programme als auch des Manuals abrufen zu können, wird ebenfalls sich der Game Logic bedient.

In der start-Datei werden die menu- sowie die manual-Szenen wie in Abbildung 97 bis 101, realisiert.

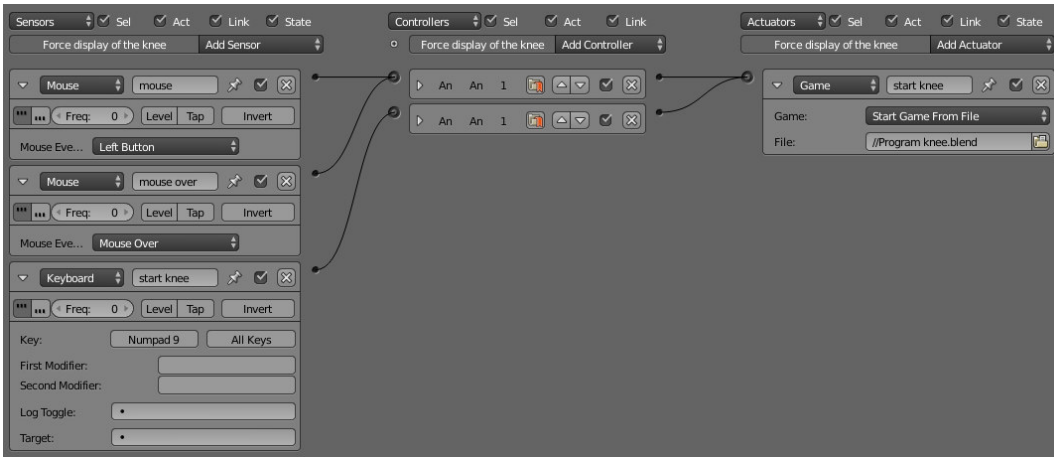


Abb. 97: Game Logic – menu-Szene: menu – Force display of the knee

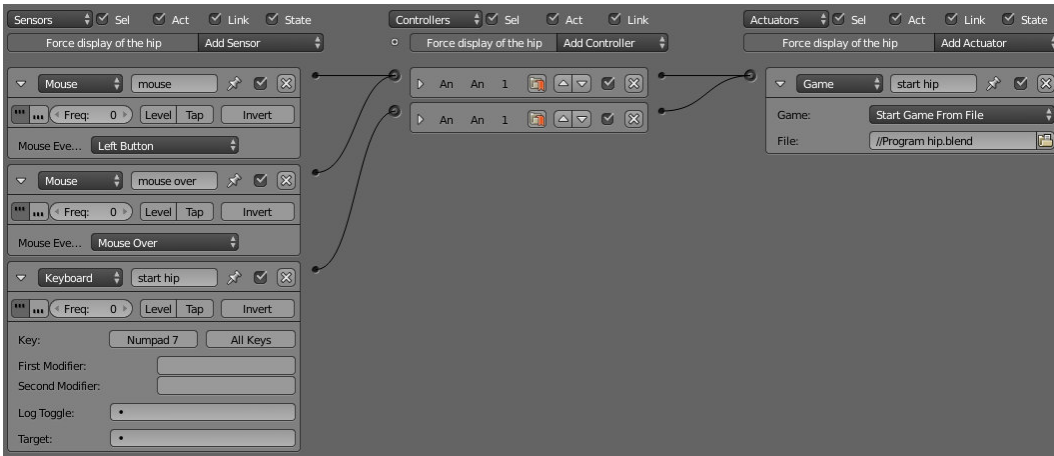


Abb. 98: Game Logic – menu-Szene: menu – Force display of the hip

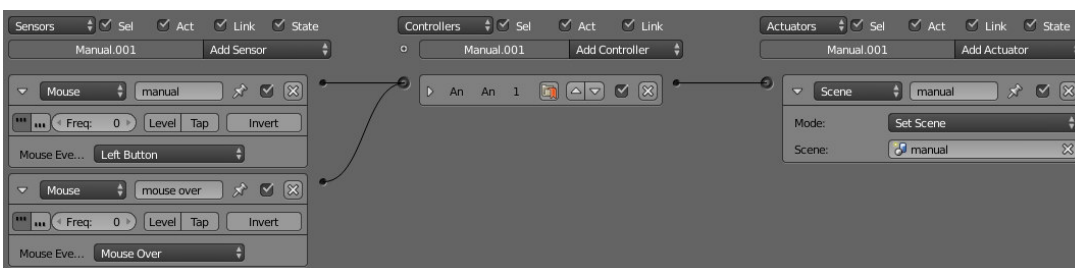


Abb. 99: Game Logic – menu-Szene: menu – Manual

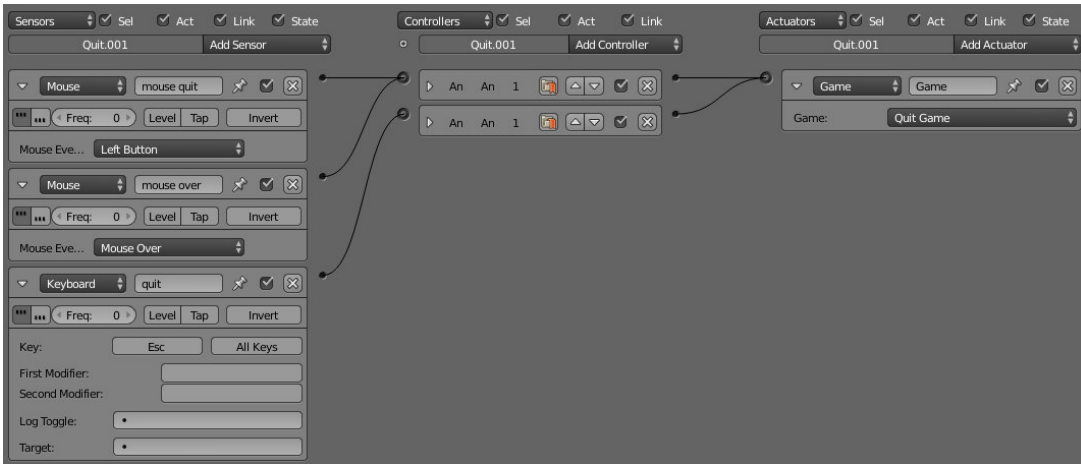


Abb. 100: Game Logic – menu-Szene: menu – Quit

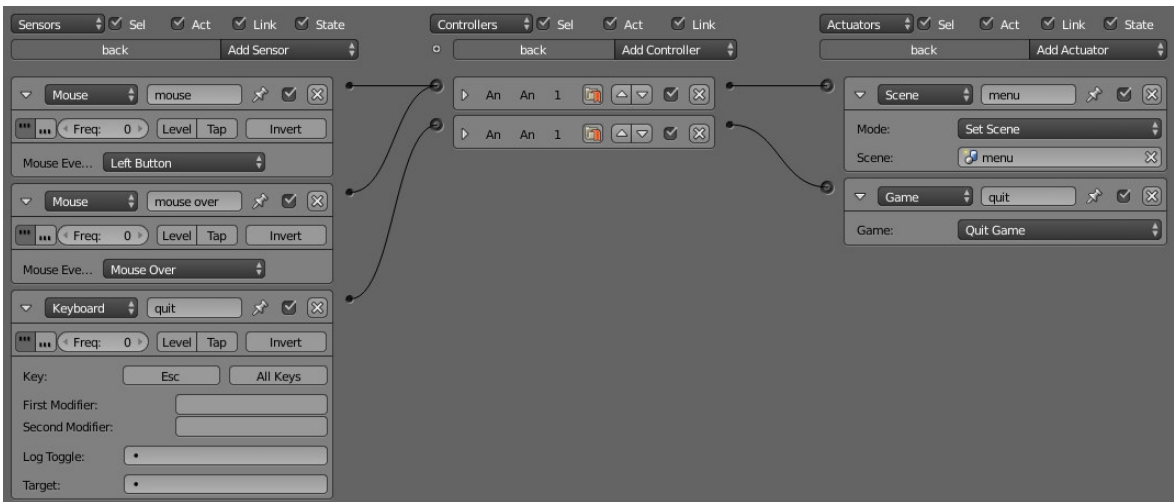


Abb. 101: Game Logic – manual-Szene: manual – back

Um zwischendurch die Funktionalität der erstellten Befehle überprüfen zu können, wird im Object Mode mittels **[P]** in die Visualisierungsansicht gewechselt.

21 Knee-Datei

Die bisherige Konstruktion für die Visualisierung der Bewegungsabläufe samt Kraftanzeige wurde in der hip-Datei verwirklicht. Diese Datei wird nun geöffnet und als knee-Datei abgespeichert. So muss die knee-Datei nicht von Anfang an neu konstruiert werden, da die bisherigen erstellten Eigenschaften auch für das Kniegelenk gelten. In dieser Datei werden für das Knie relevante Einstellungen vorgenommen:

- ⇒ die Beschriftung der Kraftbalken
- ⇒ Kraftbalkenausrichtung (Tab. 6)
- ⇒ zoom cam wird in der Höhe zum Kniegelenk verschoben und an Femur geparentet

Die Ausrichtung der Kraftanzeigen am Kniegelenk während des Gehens erfolgt entsprechend In-Vivo-Messungen von OrthoLoad und wird wie in Tabelle 6 animiert (<http://www.orthoload.com/database/> 2016).

<i>Gangphase</i>	<i>Frame</i>	<i>Krafteinwirkung</i>	<i>Frame</i>
	<i>Referenzbein</i>	<i>[% BW]</i>	<i>kontralaterales Bein</i>
Initial contact		102	
walk	1		21
slow motion	1		61
Loading response		233	
walk	4		25
slow motion	10		72
Mid stance			
mid Mid stance¹⁸	10	197	32

¹⁸ In der mid Mid stance interpoliert Blender den Wert der Kraftanzeige.

späte Mid stance		302	
walk	16		35
slow motion	46		102
Terminal stance		114	
walk	21		37
slow motion	61		108
Pre-swing		27	
walk	25		4
slow motion	72		10
Initial swing		19	
walk	31		6
slow motion	85		16
Mid swing		66	
walk	35		9
slow motion	102		25
Terminal swing¹⁹		88	
walk	38		-
slow motion	111		-

Tabelle 6: Ausrichtung der Kraftbalken am Kniegelenk, walk, slow motion

Die Kraftanzeige für stand sowie one leg stand wird ebenfalls entsprechend der oben genannten In-Vivo-Messungen mit einer angezeigten Kraft von 101% BW sowie 285% BW umgesetzt (ebd.). In der Kniebeuge wirkt eine maximale Kraft von 560% BW auf das Kniegelenk (Mow 1991, S. 47).

¹⁹ Auf der kontralateralen Seiten liegen die Frames von Terminal swing sowie Initial contact sehr nah bei einander. An dieser Stelle wird die angezeigte Kraft mittels der Software interpoliert.

22 Programmerstellung

Insgesamt sind drei Blender-Datei erstellt worden: hip-, knee- sowie start-Datei. Diese befinden sich in einem gemeinsamen Ordner. Um die Funktionsfähigkeit des Programms „Motion and force display“ an anderen Computern ohne einer installierten Blender-Software zu gewährleisten, müssen alle drei Blender-Dateien relativiert werden.

Diese Einstellung wird in File unter External Data vorgenommen (Abb. 102).

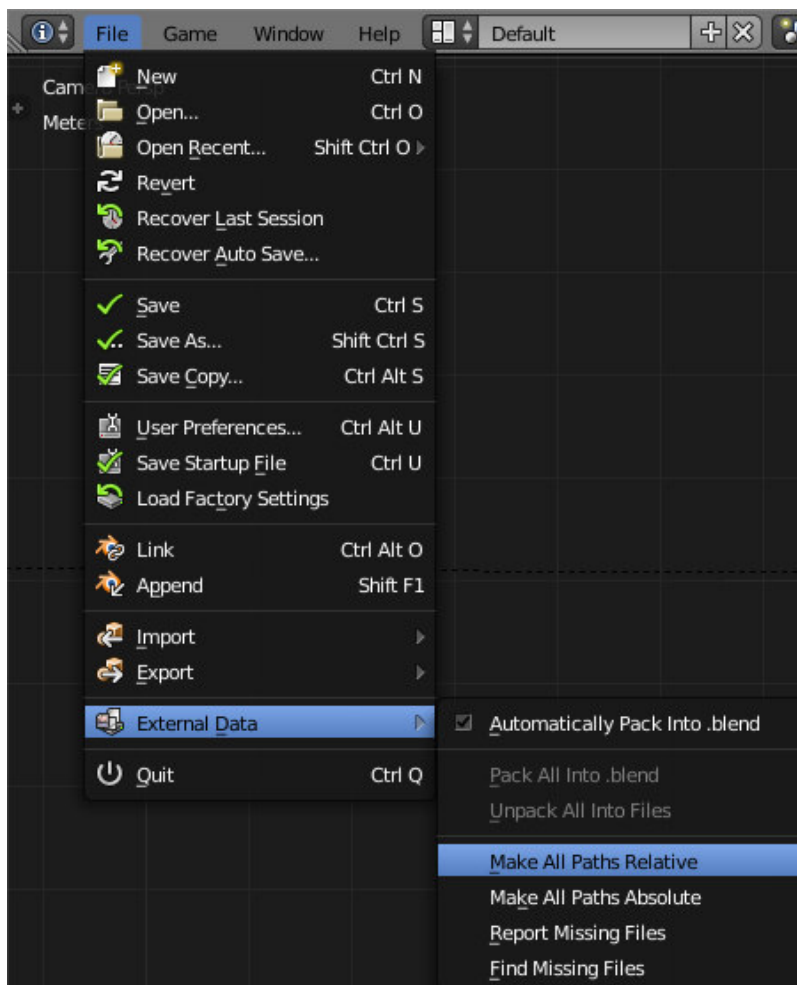


Abb. 102: Make All Paths Relative

Das abgeschlossene Programm soll vom Menü aus gestartet werden. Es folgt eine Erstellung der ausführenden exe-Datei aus der start-Datei aus. Als erstes wird dafür unter User Preferences in den Add-ons wird Game Engine: Save As Game Engine Runtime aktiviert (Abb. 103).

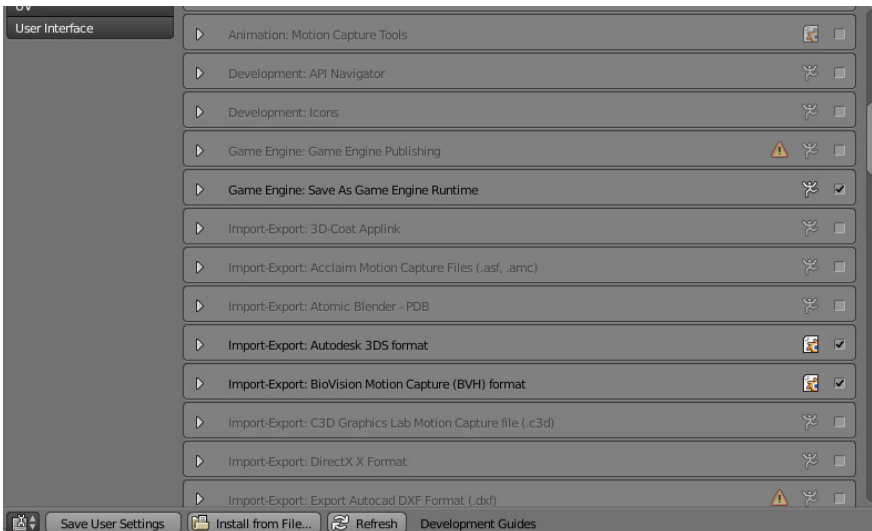


Abb. 103: Game Engine: Save As Game Engine Runtime – Aktivierung

Die Fertigstellung des Programms wird nun in File unter Export und Save As Game Engine Runtime verwirklicht (Abb. 104).

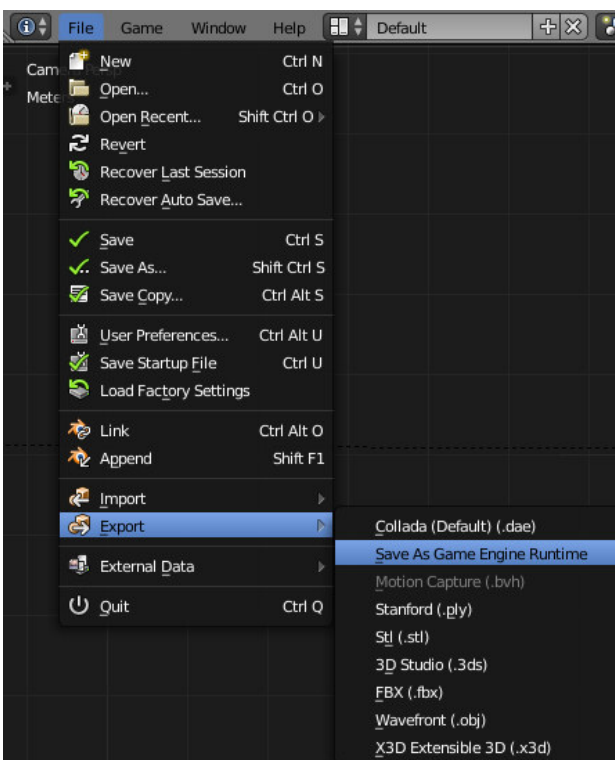


Abb. 104: Fertigstellung des Programms

Das Programm „Motion and force display“ wird mit dieser .exe-Datei geöffnet.

Abbildungsverzeichnis

Abb. 1: Arbeitsoberfläche von Blender	4
Abb. 2: frontal-zentrierte Positionierung des Skeletts	5
Abb. 3: gemeinsamer Orientierungspunkt der Skelettknochen	5
Abb. 4: Skalierung des Skeletts	6
Abb. 5: Benennung der Join-Elemente im Outliner	8
Abb. 6: Bone	9
Abb. 7: Snap - Fenster	9
Abb. 8: Erstellung eines Bones	10
Abb. 9: Hilfs-Bones zur Erstellung des Sacrum-Bones	11
Abb. 10: Armature an der Wirbelsäule und Kopf	12
Abb. 11: Clear Parent – Fenster	12
Abb. 12: Make Parent – Fenster	13
Abb. 13: Armature am Oberkörper, Benennung der Bones	15
Abb. 14: Armature am Kniegelenk	16
Abb. 15: Viewport Shading – Fenster	16
Abb. 16: Set Parent To – Fenster	18
Abb. 17: Deform – Deaktivierung	20
Abb. 18: CTRL- und IK-Bones in Verbindung mit dem Haupt-CTRL-Bone	21
Abb. 19: Erstellung des Copy Rotation – Constraints	22
Abb. 20: unvollständiger IK-Constraint am MC-Bone	23
Abb. 21: Add Constraint (with Targets) – Fenster	24
Abb. 22: Add IK – Fenster	25
Abb. 23: IK-Constraints an Femora und Tibie	26
Abb. 24: Ebene mit IK-Constraints	27

Abb. 25: Bones untergliedert nach Bones Groups, Farben und in Stick-Form	28
Abb. 26: Erstellung eines Kreises für Custom Shape Funktion	29
Abb. 27: Properties Editor: Icon Object	30
Abb. 28: Custom Shapse am Mesh	30
Abb. 29: Ebene für den Custom Shape für den Haupt-CTRL-Bone	30
Abb. 30: I Hauptkreiserstellung des Custom Shapes für den Haupt-CTRL-Bone	31
Abb. 31: II Hauptkreiserstellung des Custom Shapes für den Haupt-CTRL-Bone	32
Abb. 32: fertig erstelltes Custom Shape für den Haupt-CTRL-Bone	33
Abb. 33: Mesh mit Custom Shapes	33
Abb. 34: Seitenansicht; Voreinstellung für die Animation	35
Abb. 35: Animation, stand	36
Abb. 36: Animation walk, Frame 1, Initial contact	37
Abb. 37: Animation walk, Frame 4, Loading response	38
Abb. 38: Animation walk, Frame 10, mid Mid stance	39
Abb. 39: Animation walk, Frame 16, late Mid stance	39
Abb. 40: Animation walk, Frame 21, Terminal stance	40
Abb. 41: Animation walk, Frame 25, Pre-Swing	40
Abb. 42: Animation walk, Frame 31, Initial swing	41
Abb. 43: Animation walk, Frame 35, Mid swing	41
Abb. 44: Animation walk, Frame 38, Terminal swing	42
Abb. 45: vollendete Animation des Gangzyklus	43
Abb. 46: Animation one leg stand, Frame 33	44
Abb. 47: Animation to crouch down, Frame 35	44
Abb. 48: Animation Frame 1, slow motion	45
Abb. 49: Farbabstufung für den Hintergrund	46
Abb. 50: Aktivierung des World Background	46

Abb. 51: Erstellung des Virtuellen Bodens	47
Abb. 52: Positionierung der full body cam	48
Abb. 53: Aus der Kameraperspektive: full body cam	48
Abb. 54: Positionierung der zoom cam	49
Abb. 55: Aus der Kameraperspektive: zoom cam	49
Abb. 56: Positionierung der Lichtquellen	50
Abb. 57: New Scene	51
Abb. 58: Kamerapositionierung: Kraftsäulen	51
Abb. 59: Kraftsäulen: underground plane	52
Abb. 60: Farbgebung der Kraftsäulen: upper plane	53
Abb. 61: Kraftbalken: Textpositionierung	53
Abb. 62: Kraftsäulen inklusive Beschriftung und overlay plane	54
Abb. 63: Break, Kameraperspektive	55
Abb. 64: Animation: Kraftanzeige vom Hüftgelenk, stand	56
Abb. 65: Animation: Kraftanzeige vom Hüftgelenk, Frame 1, walk	59
Abb. 66: Animation: Kraftanzeige vom Hüftgelenk, Frame 140, one leg stand	59
Abb. 67: Menü-Ansicht	60
Abb. 68: Menü-Ansicht: Buttons	60
Abb. 69: Manual-Ansicht	61
Abb. 70: Manual-Ansicht: Button	61
Abb. 71: start-Datei: Positionierung der Kamera sowie der Lichtquellen	62
Abb. 72: File – Fenster: User Preferences	62
Abb. 73: User Preferences – Fenster: Add-ons	63
Abb. 74: Add-ons – Fenster: Import-Export: Import Images as Planes	63
Abb. 75: Import: Menü-Ansicht	64
Abb. 76: menu-Ansicht	64

Abb. 77: manual-Ansicht	65
Abb. 78: Text Editor: Window Header	65
Abb. 79: mouse on: Mausanzeige aktiv	66
Abb. 80: Logic Editor	66
Abb. 81: Game Logic – world-Szene: menu	67
Abb. 82: Game Logic – world-Szene: quit	68
Abb. 83: Game Logic – world-Szene: rotation left	68
Abb. 84: Game Logic – world-Szene: rotation right	69
Abb. 85: Game Logic – world-Szene: Wechsel zum Knieprogramm	69
Abb. 86: Game Logic – world-Szene: one leg – Action-Typ	70
Abb. 87: Game Logic – world-Szene: one leg stand gesamt	71
Abb. 88: Game Logic – world-Szene: walk	71
Abb. 89: Game Logic – world-Szene: slow motion	72
Abb. 90: Game Logic – world-Szene: to crouch down	72
Abb. 91: Game Logic – world-Szene: stand	73
Abb. 92: Game Logic – overlay-Szene: rechte Kraftsäule, one leg stand	74
Abb. 93 Game Logic – overlay-Szene: rechte Kraftsäule, gesamt	74
Abb. 94: Game Logic – overlay-Szene: linke Kraftsäule	75
Abb. 95: Game Logic – overlay-Szene: overlay plane	75
Abb. 96: Game Logic – break-Szene: break	76
Abb. 97: Game Logic – menu-Szene: menu – Force display of the knee	77
Abb. 98: Game Logic – menu-Szene: menu – Force display of the hip	77
Abb. 99: Game Logic – menu-Szene: menu – Manual	77
Abb. 100: Game Logic – menu-Szene: menu – Quit	78
Abb. 101: Game Logic – manual-Szene: manual – back	78
Abb. 102: Make All Paths Relative	81

Abb. 103: Game Engine: Save As Game Engine Runtime – Aktivierung	82
Abb. 104: Fertigstellung des Programms	82

Tabellenverzeichnis

Tabelle 1: Join-Elemente der Skelettknochen	8
Tabelle 2: Join-Elemente des Oberkörpers	14
Tabelle 3: Join-Elemente der unteren Extremitäten	17
Tabelle 4: Einteilung der Gangphasen zu den entsprechenden Frames	38
Tabelle 5: Ausrichtung der Kraftbalken am Hüftgelenk, walk, slow motion	58
Tabelle 6: Ausrichtung der Kraftbalken am Kniegelenk, walk, slow motion	80

Literaturverzeichnis

Beck, Th. *Blender 2.7. Das umfassende Handbuch*. Bonn: Galileo Press, 2015.

Bergmann, G. *Biomechanik*. Bde. Becken, Hüfte, Tschauner Ch. (Hrsg.), in *Orthopädie und Orthopädische Chirurgie. Das Standardwerk für Klinik und Praxis*, Herausgeber: Wirth C. J. Zichner L., 17-28. Stuttgart: Thieme, 2004.

Bergmann, G., et al. „Hip contact forces and gait patterns from routine activities.“ *Journal of Biomechanics* 34, 2001: 859-871.

Götz-Neumann, K. *Gehen verstehen. Ganganalyse in der Physiotherapie*. Aufl. 3. Stuttgart: Thieme, 2011.

Mow, V., Hayes, W. C. *Basic Orthopaedics*. New York: Lippincott-Raven, 1991.

Schünke M. et al. *PROMETHEUS. LernAtlas der Anatomie. Allgemeine Anatomie und Bewegungssystem*. 2. Aufl. Stuttgart: Thieme, 2007.

Wartmann, C. *Das Blender-Buch. 3D-Graphik und Animation mit Blender 2.5*. 4. Aufl. Heidelberg: dpunkt, 2011.

<http://www.orthoload.com/database/>. 26. 03. 2016.

SC Computer Graphics Studio Srl. *Human Skeleton 3d model*. 23. 05. 2015.
<https://www.cgstud.io/3d-model/skeleton-26336>.

Lehrprogramm für verwendete Blenderfunktionen

König, S. *Blender 2.7 - Das umfassende Training*. Bonn: Rheinwerk Verlag, 2014

Internetverzeichnis für verwendete Blenderfunktionen

Das Forum [blend.polis](https://www.blendpolis.de) wurde bei den Konstruktionsvorgängen zu Rate gezogen:

<https://www.blendpolis.de>

⇒ Animation

<http://cgi.tutsplus.com/tutorials/create-an-animation-walk-cycle-in-blender-using-rigify--cg-17812>

<https://www.youtube.com/watch?v=HPeqFtTQ2BM>

<https://www.youtube.com/watch?gl=DE&v=th7phr7p8RU&hl=de>

⇒ Game Engine

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

<https://www.blendpolis.de/viewtopic.php?t=39718&f=44>

<https://www.youtube.com/watch?hl=de&v=HPeqFtTQ2BM&gl=DE>

⇒ Kamera

<https://www.youtube.com/watch?gl=DE&v=5bLpPKeztkU&hl=de>

⇒ Kraftbalkendiagramm

<https://www.youtube.com/watch?v=KDDsg9Wm8-w>

⇒ Rigger

<https://www.youtube.com/watch?v=jFOERj1y7-Q>

<https://www.youtube.com/watch?v=mrcWEGb2gw8>

<https://www.youtube.com/watch?v=O1mLUwj-pVk>

⇒ Startmenü

https://www.youtube.com/watch?v=KeUgOPOt_94

⇒ Pausefunktion

<https://www.youtube.com/watch?v=clhCwwRBo-8>

<https://www.youtube.com/watch?gl=DE&v=3NWqN2kl9Jc&hl=de>