

Animation 2 – „Rigging“  
SS 2021

**Lehreinheit 1 & 2**

22.04.2021 / 29.04.2021





# Bastian Kramer

## 3D Artist | Interactive Designer

- BA Mediendesign  
MA Design und Medien  
Hochschule Hannover
- Character und Pipeline TD
- Rigging, Cloth- und Hair-Simulation
- > 8 Jahre Erfahrung
- 5 Full CG Filme
- Maya, Unity 3D, UnrealEngine  
C++, C#, Python
- [www.kramerbastian.de](http://www.kramerbastian.de)
- [mail@kramerbastian.de](mailto:mail@kramerbastian.de)



Quelle: Ambient Entertainment



Quelle: Ambient Entertainment

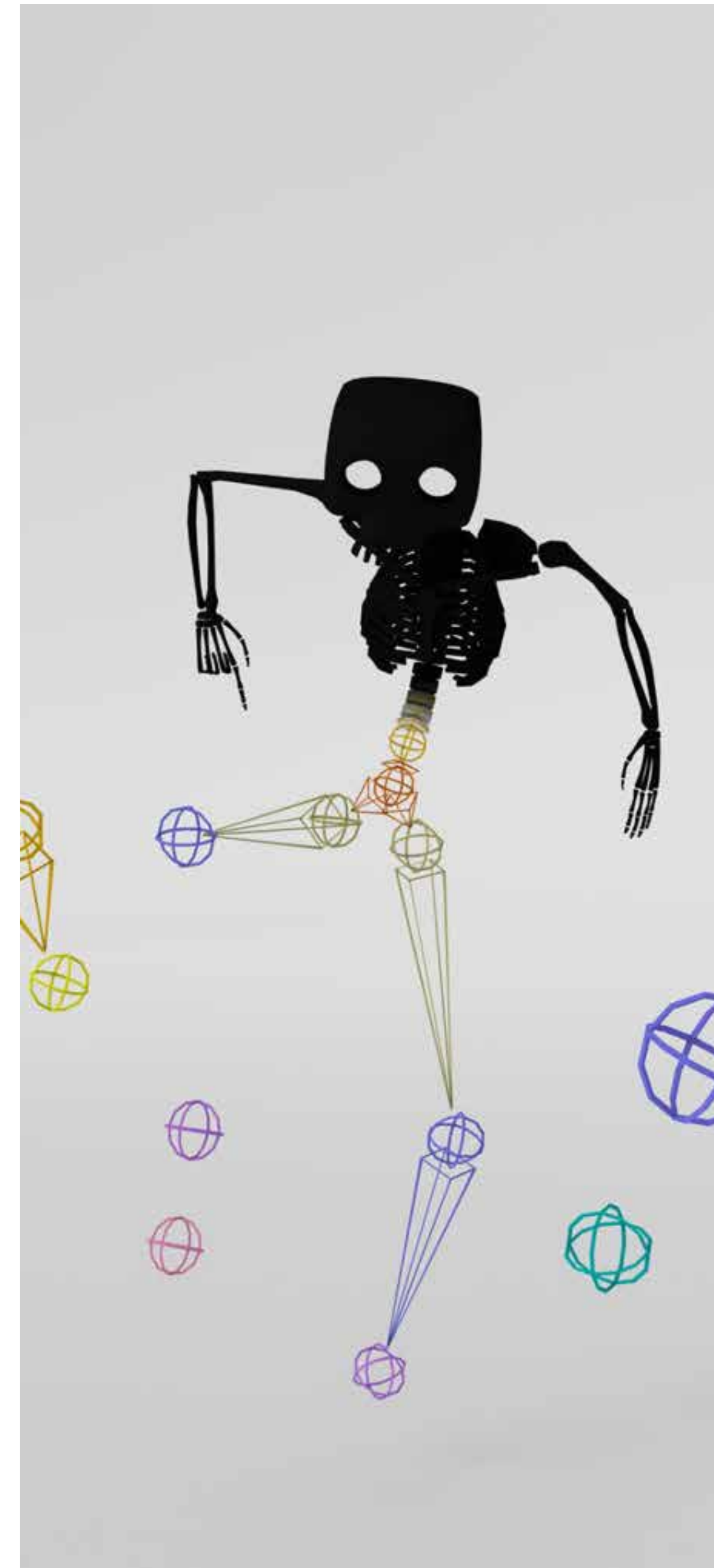


Quelle: Ambient Entertainment



# Kursinhalt

- Einführung in die Grundlagen des Riggens
- Ablauf und Struktur beim Aufsetzen eines Rigs
- Kurze Vorstellung einiger Rigging Tools
- Aufsetzen eines Character-Rigs in groben Zügen



# Übersicht:

- Was ist Rigging?
- Was sind die Aufgaben eines Riggers?
- Einordnung in die Produktion
- Fachbegriffe
- Vor dem Riggen...
- Aufgaben und Schritte
- Rigging Tools
- Struktur und Benennung
- Fragerunde

# Was ist Rigging?





# Was ist Rigging?

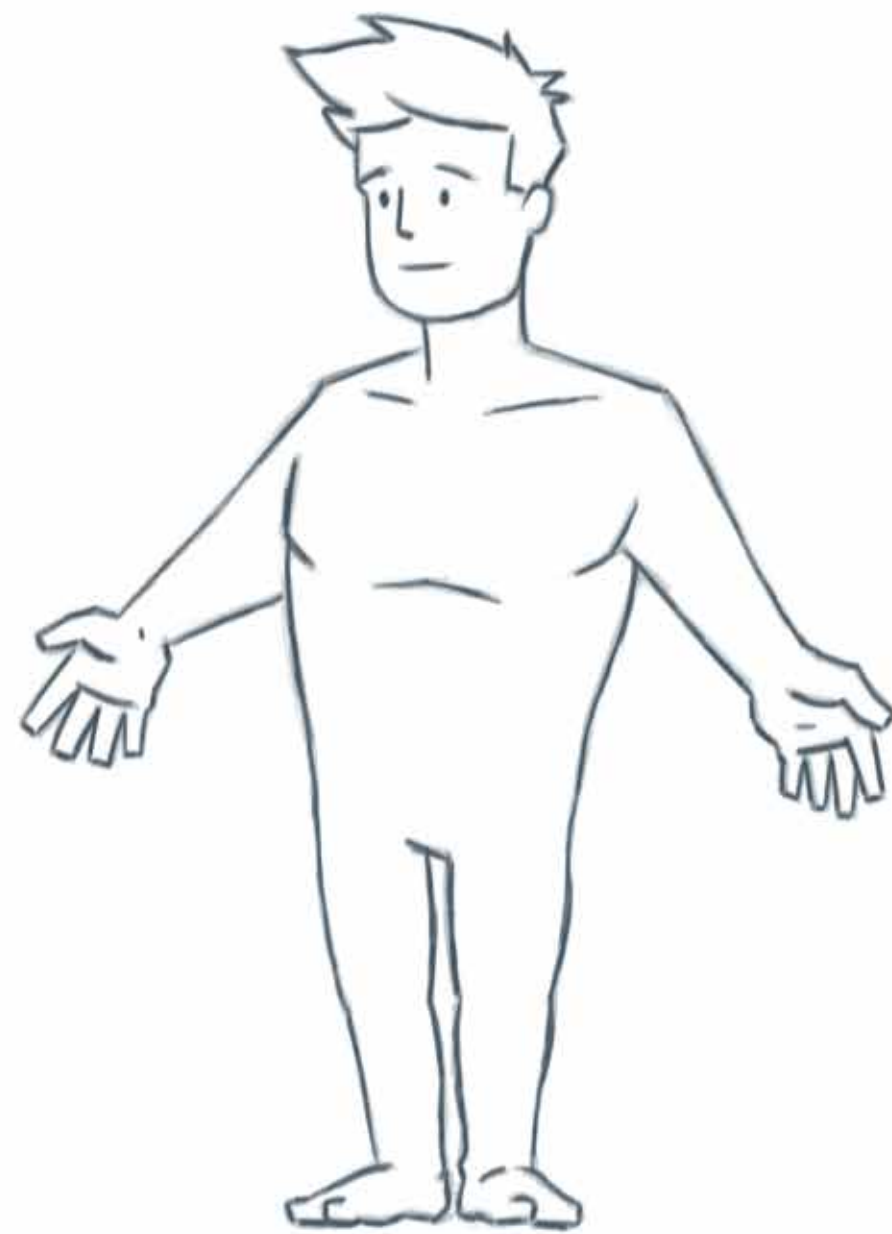
- Bindeglied zwischen Modeling und Animation
- Einsetzen der Mechanik und der Funktionalität
- Definiert den Rahmen der Bewegung
- Meist verbunden mit Pipeline:  
Asset Management  
Asset Checking  
Export, Import von Caches  
Simulation Setups



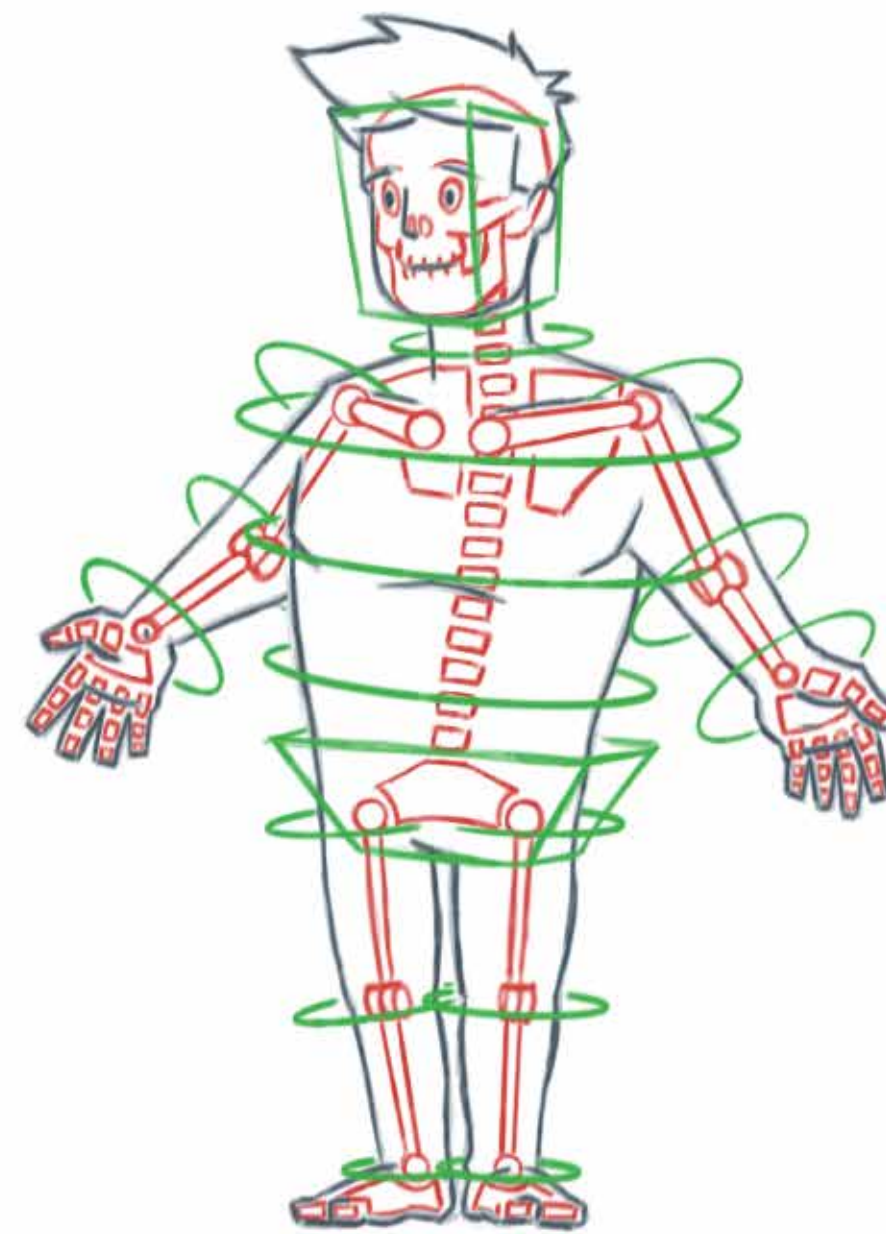


# Was ist Rigging?

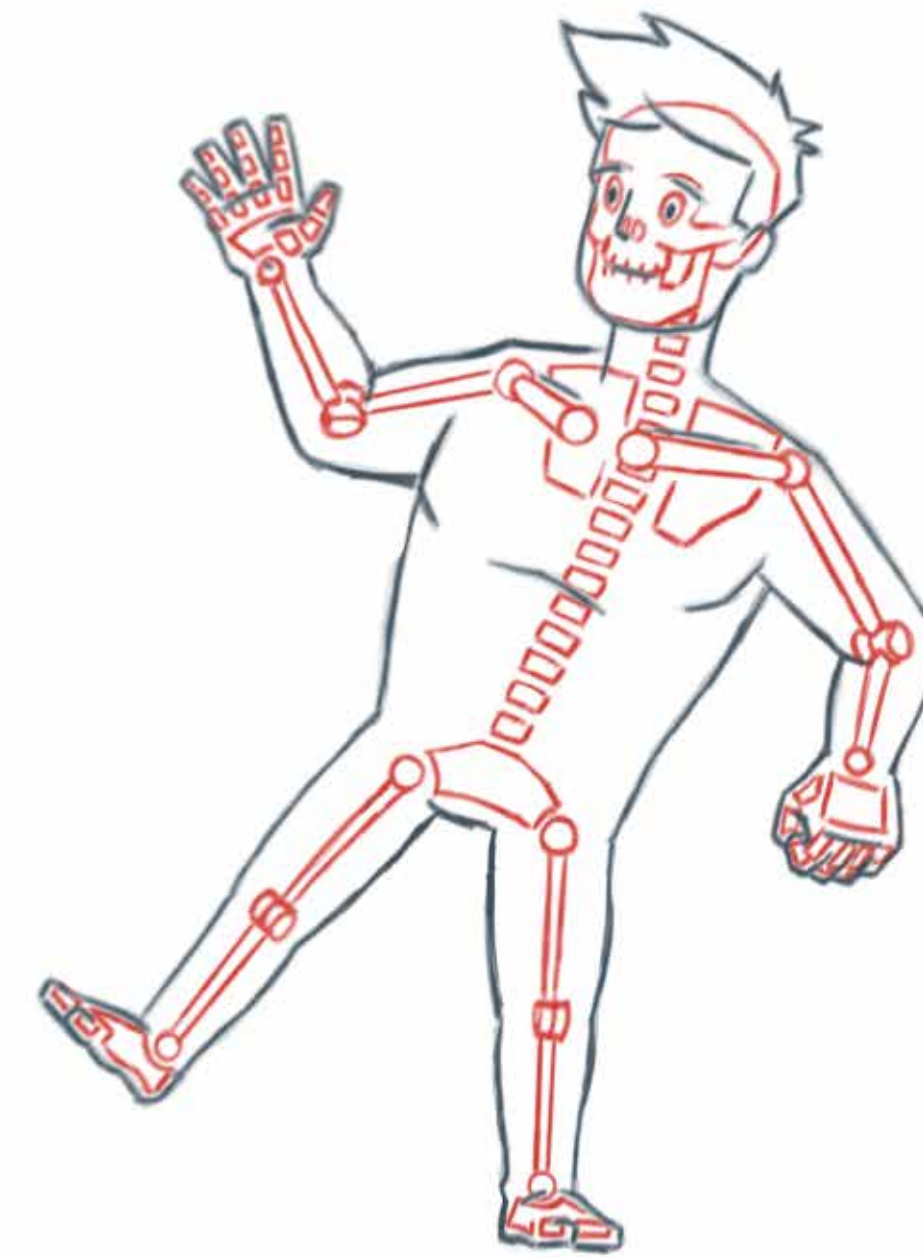
Modeling



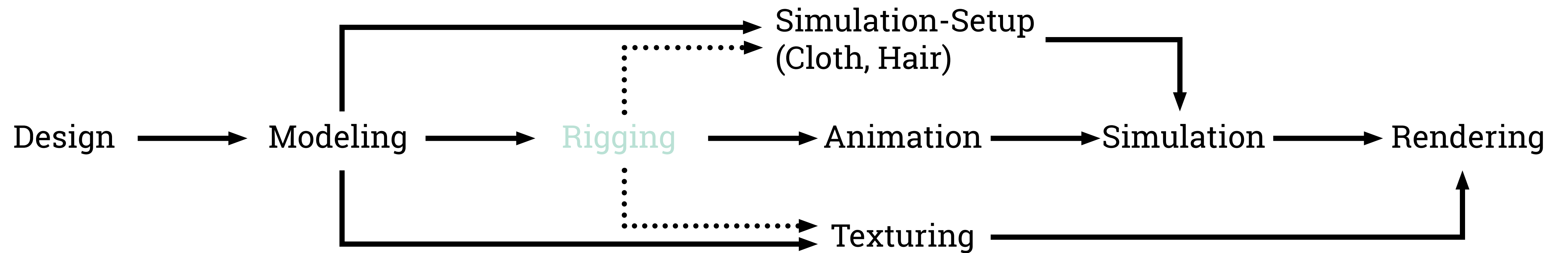
Rigging



Animation



# Einordnung in die Produktion





# Fachbegriffe

## Rig (Setup)

Oberbegriff für alle Formen von Mechanik in einem 3D Objekt

## Body-Rig

Rig für Körper-Funktionen

## Facial-Rig

Rig für Gesichts-Funktionen

## Joints / Bones

Gelenke / Knochen. Basis für die skeletale Deformation eines 3D Objekts

## Skelett

(Teil-)Hierarchische Anordnung von Joints meist unter einem Root

## Root

Oberster Joint in einer Skelett-Hierarchie

## Transform

Oberbegriff für alle Elemente mit räumlichen Koordinaten

## Controls

Kontroll-Elemente, die über Transformation oder Attribute das Rig steuern

## Null-Gruppen

Transformations Gruppen zum „ausnullen“ untergeordneter Elemente

## Locator

Visueller Transform zur Konstruktion von Mechaniken

## Constraint

„Einschränkung“ oder Abhängigkeit zwischen verschiedenen Transforms

## Deformer

Verformung von Shapes / Geometrie mittels Attributen oder anderen Elementen

## Skinning

Gewichtete Deformation eines Shapes durch Joints

## Weightpainting

Gewichtungen des Skinings definieren

## Initialpose / Bindpose

Ausgangspose des Models bzw. des Skinings

## T-Pose

Initialpose mit Armen im 90° Winkel

## A-Pose

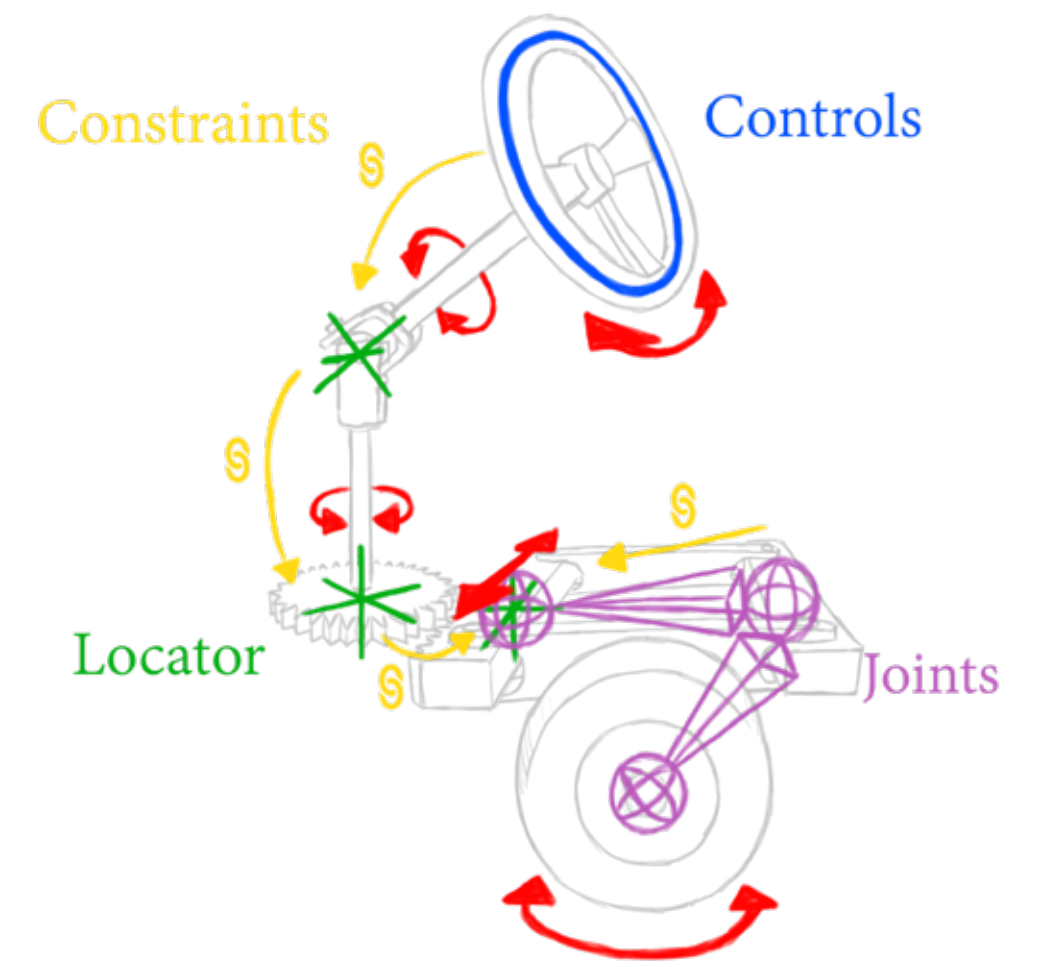
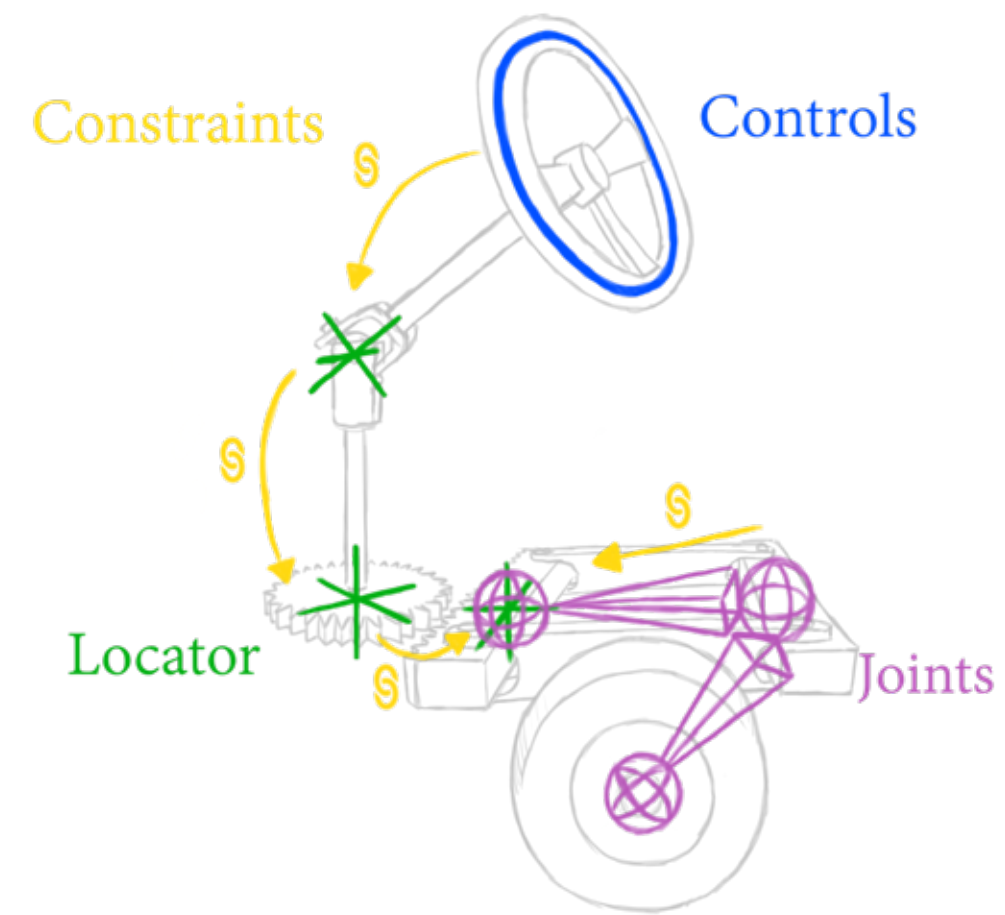
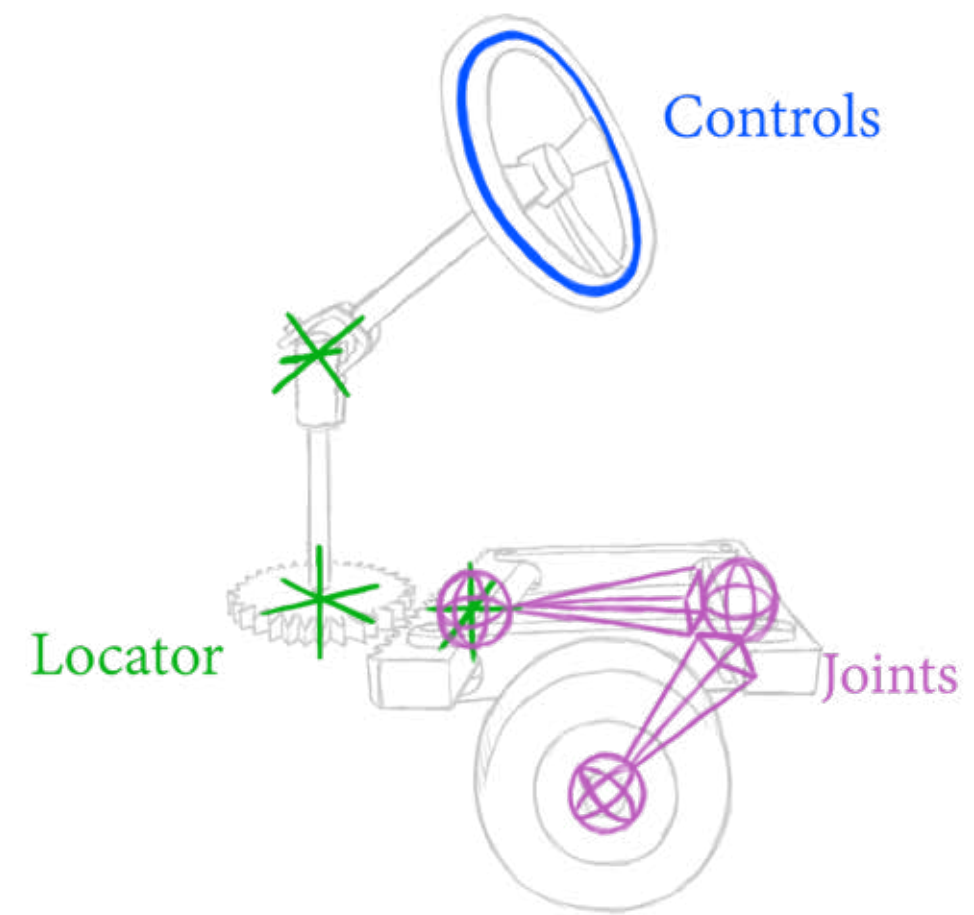
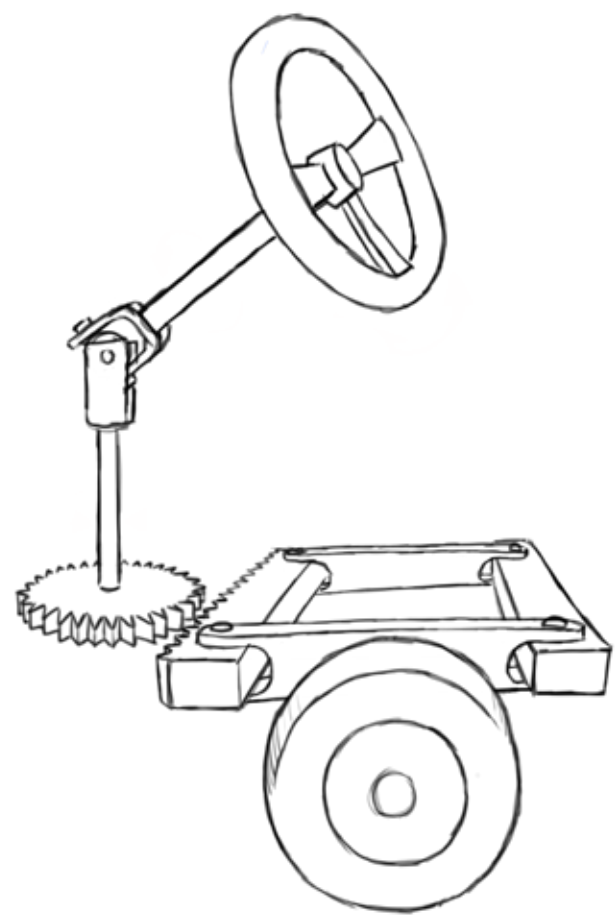
Initialpose mit Armen im 45° Winkel

## FK (Forward Kinematic)

Hierarchisch Abfolgende Transformation. Die Drehungen der Gelenke bestimmen die Pose

## IK (Inverse Kinematic)

Umgekehrt Abfolgende Transformation. Die Pose bestimmt die Drehungen der Gelenke





# Vor dem Riggen

## Design

- Gelenke deutlich machen
- Deutliche Übergänge zwischen Körper und Gliedmaße
- Mechanische Elemente sollten entsprechenden Freiraum haben
- Extra Elemente bedürfen extra Setups
- Körpervolumina beachten
- Anatomie und Proportionen
- 2D funzt nicht immer in 3D!
- Scale Sheet

# Vor dem Riggen

## Modeling

- Edgeflow beachten. Falten und Gelenke nachbilden
- Auflösung der nötigen Deformation anpassen
- Triangles vermeiden, keine N-Gons mit mehr als vier Seiten
- Edgeloops um Gesichts- und Gelenkpartien
- Faces möglichst eben halten
- Zu viel Detail im Mesh kann das Weighting erschweren
- In original Größe modellieren
- Keine Intersections



# Aufgaben und Schritte

1. **Klärung der Zielsetzung:**  
Was soll das Rig können?  
Welche Aufgaben soll es erfüllen?
2. **Analyse des Models:**  
Wo sitzen Gelenke?  
Wo brauche ich Deformation?  
Wo brauche ich Controlls?  
Ist das Modell geeignet?
3. **Aufsetzen der initialen Pose:**  
Position und Ausrichtung der Gelenke festlegen
4. **Aufsetzen des Skeletts:**  
Joints hierarchisch anordnen, platzieren und ausrichten
5. **Aufsetzen der Controlls:**  
Wo brauche ich Dreh- und Translations-Punkte?  
Wie steuere ich das Objekt / Teilbereiche?  
FK und oder IK?  
Brauche ich weitere Attribute?
6. **Binden des Skeletts an die Controlls:**  
Constraints knüpfen.  
Mechaniken konstruieren.
7. **Deformation aufsetzen:**  
Skinning anwenden und weightpainten.  
Blendshapes und zusätzliche Deformer
8. **Aufräumen:**  
Alles löschen, was nicht zum Rig gehört!

```

event.py - PyCharm Community Edition 2017.2.4
class EventDispatcher(object):
    def __init__(self, listeners = None):
        self._listeners = listeners if isinstance(listeners, list) else []

    def __del__(self):
        self.unregisterAll()

    def __call__(self, *args, **kwargs):
        self.invoke(*args, **kwargs)

    def register(self, listener):
        if listener not in self._listeners:
            self._listeners.append(listener)

    def unregister(self, listener):
        self._listeners.remove(listener)

    def unregisterAll(self):
        self._listeners = []

    def hasListeners(self):
        return len(self._listeners) > 0

    def invoke(self, *args, **kwargs):
        for listener in self._listeners:
            listener(*args, **kwargs)

if name == " main ":
    class Foo(object):
        def __init__(self):
            self._onA = EventDispatcher()
            self._onB = EventDispatcher()
            self._onC = EventDispatcher()

        @property
        def onA(self):
            return self._onA

        @property
        def onB(self):
            return self._onB

        @property
        def onC(self):
            return self._onC

        def a(self):
            self._onA.invoke("a")

        def b(self):
            self._onB.invoke("b")

        def c(self):
            self._onC.invoke("c")

        def do(self):
            self.a()
            self.b()
            self.c()

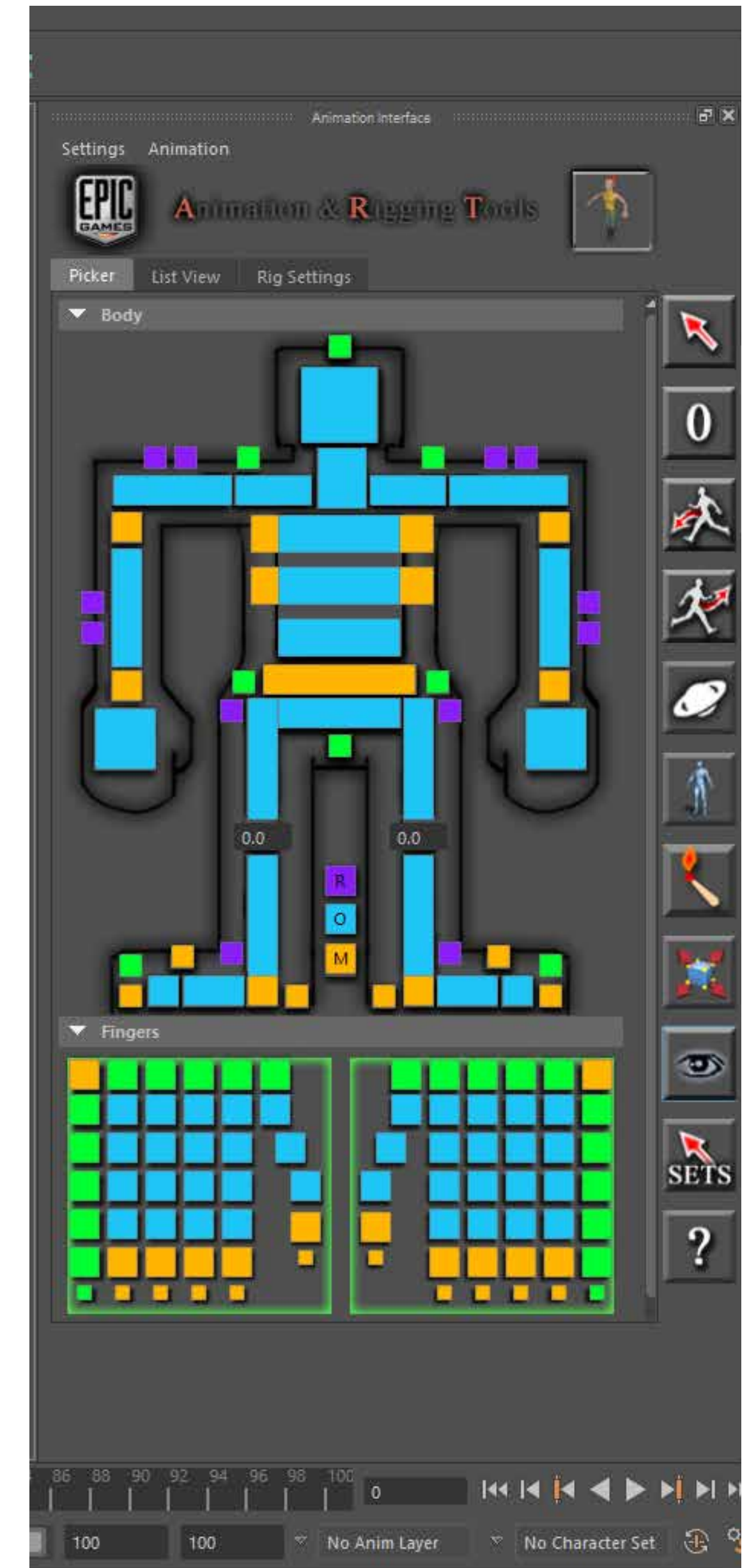
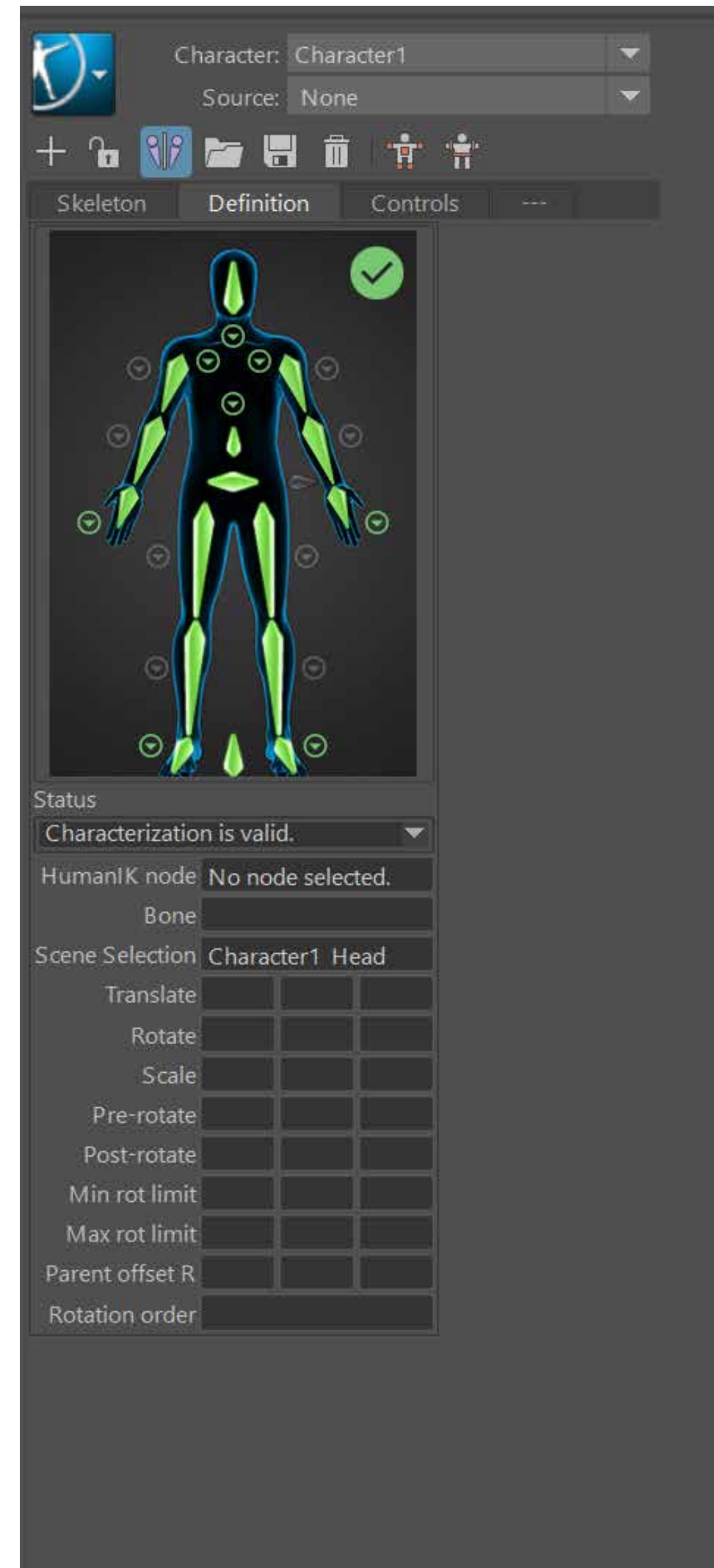
        def test(arg):
            print arg

    foo = Foo()

```

# Rigging Tools und Systeme

- MEL / Python
- Plugins
- Maya Quick-Rig / Human-IK
- mGear
- Advanced Skeleton
- ART Tools von Unreal
- Mixamo von Adobe





# Struktur und Benennung

## Struktur im Projekt

- Trennung der Prozessschritte
- Work – Final
- Arbeiten mit Referenzen

# Struktur und Benennung

## Struktur in der Datei

- Sauberes importieren / referenzieren des Models
- Elemente Gruppieren
- Klare Trennung von
  - Geometrie
  - Skelett
  - Controlls
  - Setups
- Unnötige Nodes löschen



# Struktur und Benennung

## Benennung

- Einheitliche Schreibweise
- Projekt-Übergreifend
- camelCase / PascalCase / unicasel / under\_score
- Präfix / Suffix
- Unikate
- Alle Nodes beachten
- Eindeutige Zuordnung über Namen

# Fragen?