

Prozedurale Generierung von Wirbeltierskeletten

Masterarbeit von

Nina Zimbel

an der Fakultät für Informatik
Lehrstuhl für Computergrafik

Erstgutachter:	Prof. Dr.-Ing. Carsten Dachsbacher
Zweitgutachter:	Prof. ?
Betreuender Mitarbeiter:	Dr. Johannes Schudeiske

xx. Month 20XX – xx. Month 20XX

1 Bisherige Arbeiten

1.1 Skelett der Wirbeltiere

- „Dem Skelett der Wirbeltiere sind viele Gemeinsamkeiten ansehbar, trotzdem unterscheidet es sich, je nach Lebensraum und Anforderungen, teilweise erheblich. Mit diesen Gemeinsamkeiten und Unterschieden beschäftigt sich die Vergleichende Anatomie.“ (<https://de.wikipedia.org/wiki/Skelett#Wirbeltiere>) → über vergleichende Anatomie informieren
- „Von vielen Zoologen wird heute der Begriff Schädeltiere (Craniota) für dieses Taxon bevorzugt. Diese Auffassung berücksichtigt, dass die Rundmäuler, wie auch einige andere Wirbeltiere, als Achsenskelett keine Wirbelsäule, sondern eine Chorda dorsalis haben. Doch allen Wirbeltieren gemein ist ein verknöcherner oder knorpeliger Schädel; sein Vorhandensein gehört somit zu den gemeinsam abgeleiteten Merkmalen (Synapomorphien) dieser Chordaten-Gruppe.“ (<https://de.wikipedia.org/wiki/Wirbeltiere>) → Beschränkung auf Schädeltiere mit Wirbelsäule

1.2 No Man's Sky

- Webseite [2]
- „For creatures, basic templates of creatures that exist on the Earth were created and then manipulated by the system, changing everything from height, weight, bone density, voice pitch, what it eats, and its behaviors, even creating variation within the species.“ (<https://nomanssky.fandom.com/wiki/Biology>)
- „Creatures were often generated by mixing and matching random parts from a library, and then adjusting the underlying skeleton so that the creature appeared realistic; a creature with a tiny body could not support a giant head, for example.“ (https://en.wikipedia.org/wiki/Development_of_No_Man%27s_Sky)
- Zunächst Generierung von äußerem 3D-Modell dann Anpassung der Knochen. Hier zunächst Generierung eines „biologisch sinnvollen“ Skeletts, dann Außenhülle.

2 Idee

1. Erzeugung eines Wirbeltierskeletts
2. Erzeugung von Muskeln (optional?)
3. Erzeugung von Haut (einfache Hülle um das vorhandene Modell) (optional)

2.1 Skelett

- Iterative Erzeugung eines Skeletts durch eine probabilistische kontextfreie (?) Grammatik, die so erweitert ist, dass sie nicht ein einfaches Wort erzeugt, sondern einen Baum von Zeichen. Verwendung von paramterischen L-Systemen [1] könnte sinnvoll sein.
- Der Zufall ist eingeschränkt durch Eingabeparameter oder Benutzereingabe.
- Ein Skelett besteht aus Knochen, Gelenken und deren (relativen) Positionen und Orientierungen.
- Ein Knochen ist im einfachsten Fall ein Zylinder (Strecke) mit Länge und Radius, kann aber auch eine konvexe Kurve mit einem Radius sein.
- Ein Gelenk verbindet hat keine Ausdehnung (?). Es ist das Verbindungsstück zwischen zwei oder mehr Knochen und legt fest wie die Knochen sich relativ zueinander bewegen können. Werden mehr als zwei Knochen verbunden ist es einfach eine feste Verbindung.

2.2 Muskeln

- Die „Hauptmuskeln“ verlaufen bei Wirbeltieren wahrscheinlich ähnlich, relativ zu den Knochen.
- Knochen/Gelenke bekommen Zusatzattribute für Start- und Zielpunkte der Muskeln.
- Muskeln haben eine „Dicke“ und aus Start- und Zielpunkt muss Kurve des Muskels generiert werden.
- Wie wird genau Form festgelegt? Muskeln irgendwie auf ihre „Dicke“ aufblähen + Interaktion mit vorhandenen Elementen (andere Muskeln und Knochen)

2.3 Haut

- Was für Algorithmen gibt es, die zu einem vorhandenen 3D-Modell eine Hülle mit gewissen Eigenschaften generieren?
es gibt eine solche Funktion z.B. in AutoCAD <https://knowledge.autodesk.com/de/support/autocad/learn-explore/caas/CloudHelp/cloudhelp/2016/DEU/AutoCAD-Core/files/GUID-B7F99810-765E-4E7E-ABDD-275C64147CCC-htm.html>
- Muskeln sind dafür nicht unbedingt nötig, aber wahrscheinlich hilfreich.

3 Technische Umsetzung Skelett

- Repräsentation des Zustands als Baum von einzelnen Zeichen (terminale sowie nichtterminale).
- Programmiersprache: Rust?
- Übersetzung in ein 3D-Modell
 - Welches Dateiformat? (siehe unten)
 - Geht das mit Rust oder braucht man noch einen Zwischenschritt? Oder doch besser andere Programmiersprache?
- Wie sieht Benutzereingabe aus?

3.1 Dateiformate

- Welches Format wird am meisten Verwendet? → fbx?
- Einfachstes Format (aber auch mit kleinstem Funktionsumfang) obj
- Welches Format ist für eine interaktive Anwendung sinnvoll?

3.1.1 OBJ

- Beschreibung des Formats: <https://www.fileformat.info/format/wavefrontobj/egff.htm>
- Erzeugung mit Rust: obj_exporter <https://docs.rs/obj-exporter/0.2.0/obj-exporter/index.html>

3.1.2 FBX

- Verwendung am besten über Autodesk FBX SDK für C++.
- Dokumentation: <http://help.autodesk.com/view/FBX/2019/ENU/> und <http://docs.autodesk.com/FBX/2014/ENU/FBX-SDK-Documentation/index.html>
- Es gibt auch fbxccl, eine FBX library für Rust. Ist aber relativ low level und nicht ganz offensichtlich wie zu verwenden.

Literatur

- [1] James Scott Hanan. "Parametric L-systems and Their Application to the Modelling and Visualization of Plants". AAINN83871. Diss. 1992. ISBN: 0-315-83871-X.
- [2] *No Man's Sky*. URL: <https://www.nomanssky.com/> (besucht am 08. 10. 2019).