Karlsruher Institut für Technologie

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

(TODO: Latexvorlage CG?)

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`) **(TODO: 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 verknöcherter 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 Ziva

- Ziva VFX Maya Plugin zur Erstellung von Charakteren und Simulation von biomechanischen Bewegungen `https://zivadynamics.com/`
- Charaktererstellung in Ziva beginnt mit der Modellierung des Skeletts. Knochen mit Animationen werden als Alembic-Datei gespeichert und dann in „Ziva-Knochen“ konvertiert. `https://discover.therookies.co/2019/06/01/vfx-in-9-steps/`

1.3 ZSpheres in Zbrush

- `http://docs.pixologic.com/user-guide/3d-modeling/modeling-basics/creating-meshes/zspheres/`,
  Beispielvideo: `https://www.youtube.com/watch?v=Wl0XK6ggUOA`
- Möglichkeit ein „Skelett“ aus Kugeln zu erstellen. Definiert aber eher die grobe Außenhaut mit Zusatzinformationen dazu wo die Gelenke sind.

1.4 3DS MAX Biped

- https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2019/ENU/3DSMax-Character-Animation/files/GUID-2F6BC5D1-DD45-4C2E-AC3A-D8C6E0F5DEB1-htm.html
- Möglichkeit Skelett in einen fertig modellierten Körper einzupassen.
- Skelette sind schon vorgefertigt.
- v.a. für menschliche Skelette, aber auch (limitiert) anpassbar auf Tiere

1.5 Forensik und Archäologie

- forensische Gesichstrekonstruktion ist spezialisiert auf Menschen und verwendet Zusatzinformationen wie Stockfotos von Gesichtsmerkmalen (https://en.wikipedia.org/wiki/Forensic_facial_reconstruction)
- Rekonstruktion von Tieren in der Archäologie anhand des Skeletts v.a. durch Künstler (?)

1.6 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.

2 Idee

1. Erzeugung eines Wirbeltierskeletts
2. Erzeugung von Muskeln (future work)
3. Erzeugung von Haut (future work / kurz ausprobieren)
4. Erzeugung von vielen Skelettvarianten bei Eingabe eines Skeletts (nur wenn es relativ leicht möglich ist)

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 (nötig für Extremitäten). Verwendung von paramterischen L-Systemen [1] könnte sinnvoll sein.
- Regeln wichtig, die dafür sorgen, dass das Tier am Ende auch funktional ist.
- Der Zufall ist eingeschränkt durch Eingabeparameter oder Benutzereingabe (siehe Interaktivität).
- 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 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.
- Skelett darf nicht zu abstrakt sein, da es sonst zu wenig Informationen zum konkreten Tier liefert.
- Ein detailliertes Skelett ist für Wesen sinnvoll, die es so noch nicht gibt bzw. wo man keine richtige Vorstellung davon hat wie es „funktioniert“, z.B. bei mehr Gliedmaßen.
- Köpfe sind kompliziert → Auswahl an Köpfen bereitstellen (evtl. leicht skalier-/verformbar)

2.1.1 Extremitäten

- https://de.wikipedia.org/wiki/Extremit%C3%A4tenevolution
- „Die paarigen Flossen von Fischen und Extremitäten von Tetrapoden sind insofern homologe Skelettelemente, als sie bei beiden an Schulter- und Beckengürtel ansetzen und die Extremitäten aus den paarigen Flossen evolutionär hervorgegangen sind.[3] Sie unterschieden sich jedoch im Knochenaufbau und in der Embryonalentwicklung, so dass ein evolutionärer Übergang aus den Einzelelementen schwer erklärbar ist."
  → einfach ignorieren und trotzdem (erstmal) so modellieren?

2.2 Muskeln

- Die „Hauptmuskeln" verlaufen bei Wirbeltieren wahrscheinlich ähnlich, relativ zu den Knochen. Trotzdem unterscheiden sie sich recht stark.
- 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 die genaue Form festgelegt? Muskeln irgendwie auf ihre „Dicke" aufblähen + Interaktion mit vorhandenen Elementen (andere Muskeln und Knochen) → future work

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
- Einfach nur eine Hülle mit gewissem Abstand sieht wahrscheinlich sehr unrealistisch aus. „Bony Landmarks" (Stellen an denen das Gewebe über den Knochen sehr dünn ist) könnten helfen (siehe https://www.proko.com/landmarks-of-the-human-body/) oder „bone weights"

2.4 Interaktivität

- Eine Anwendung, bei der nach Eingabe von Parametern sofort das komplette Tier generiert wird, ist weniger hilfreich als eine, bei der schrittweise Teile davon generiert werden können (und auch rückgängig gemacht werden können)
- Teile, die einem nicht gefallen, sollten geändert werden können

- Änderungen können Auswirkungen auf den Rest des Körpers haben (durch Regeln) bzw. manche Änderungen sind nicht möglich
- Könnte verwendet werden um schnell verschiedene Möglichkeiten zu testen

3 Technische Umsetzung

- 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?

3.1 Dateiformate

- Einfachstes Format (nur für die Darstellung von 3D-Objekten ohne Zusatzinformationen): obj
- Jeder Editor geht mit Muskeln und Gelenken anders um. Gibt es ein Dateiformat, das nicht speziell zu einem Editor gehört, dass Bedingungen an die Rotation von Gelenken speichern kann?
- Welches Format ist für eine interaktive Anwendung sinnvoll?
- Eigenes Format erzeugen? Dann bräuchte man Plugins um es in verschiedenen Editoren laden zu können. Viel verwendeter Editor: Houdini (kostenlos für Studenten aber nicht Open Source). Oder selbst darstellen (siehe Interaktivität).
  - Vorschlag von Jo:
  - „Memory dumps“, also direkt die structs aus dem speicher auf platte rausschreiben. Am besten wenn sie am Stueck liegen mit einem fwrite() und zurücklesen mit einem fread(). Es ist nuetzlich dazu am Anfang der Datei ein bisschen Metadaten zu speichern (magic number, version, array size etc).

3.1.1 OpenSim

- `https://simtk-confluence.stanford.edu:8443/display/OpenSim/OpenSim+Documentation`
- Open Source Software Platform für die Modellierung uns Simulation von Menschen, Tieren, etc.
  aber vor allem gedacht zur Auswertung von experimentellen Daten

- Import von .obj Dateien möglich. Außerdem zusätzliche Daten wie Winkel von Gelenken über .mot oder .sto Dateien (eigenes Format von OpenSim, siehe `https://simtk-confluence.stanford.edu:8443/display/OpenSim/Preparing+Your+Data`)
- Export in andere Dateiformate nicht möglich (?)
- für Download und Zugang zur „Community“ Account nötig
- für Windows und Mac OS (Linux Support gibt es auch, ist aber schwieriger: `https://simtk-confluence.stanford.edu:8443/display/OpenSim/Linux+Support`)

3.1.2 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`
- Reicht wahrscheinlich für die ersten Dinge aus. Finetuning wird sowieso mit anderer Software gemacht

3.1.3 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 fbxcel, eine FBX library für Rust. Ist aber relativ low level und nicht ganz offensichtlich wie zu verwenden.
- Einschränkungen für Gelenke können in FBX nicht gespeichert werden `http://docs.autodesk.com/FBX/2014/ENU/FBX-SDK-Documentation/index.html?url=cpp_ref/class_fbx_constraint.html,topicNumber=cpp_ref_class_fbx_constraint_htmlc57a3f99-513a-44a0-a24f-445e9077c99f`

3.1.4 Alembic

- `www.alembic.io`
- Wird u.a. dafür verwendet Knochen (+ Animationen) in Ziva zu importieren
- Es kann mit Python (PyAlembic) und C++ verwendet werden.
  PyAlembic Doku: `http://docs.alembic.io/python/examples.html#pyalembic-intro`
  C++ API Refernce (enthält sehr wenig Infos): `http://docs.alembic.io/reference/index.html`
- Für Rust gibt es keine Bibliothek (?)

3.2 Interaktivität

- Verwendung von imgui (opengl/vulcan/3D view integriert): `https://github.com/ocornut/imgui` **(TODO: in Imgui einarbeiten)**
- OpenGL und Imgui für Rust: `https://nercury.github.io/rust/opengl/tutorial/2018/02/08/opengl-in-rust-from-scratch-00-setup.html`, `https://github.com/michaelfairley/rust-imgui-sdl2` **(TODO: taugt das was?)**
- Imgui Tutorials `http://headerphile.com/sdl2/opengl-part-1-sdl-opengl-awesome/`, `http://www.sdltutorials.com/sdl-opengl-tutorial-basics` **(TODO: anschauen)**

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).

[3] Lennart Olsson und Uwe Hoßfeld. “Homology, Genes, and Evolutionary Innovation.—Günter P. Wagner.” In: *Systematic Biology* 64.2 (Dez. 2014), S. 365–367. ISSN: 1063-5157. DOI: 10.1093/sysbio/syu127. eprint: http://oup.prod.sis.lan/sysbio/article-pdf/64/2/365/24587311/syu127.pdf. URL: https://doi.org/10.1093/sysbio/syu127.