

Vertex Hacking

Kleine Anleitung zum Sezieren von 3D-Dateiformaten

Markus Dahms <mad@automagically.de>

BraLUG e.V.

29. Dezember 2008

Überblick

- 1 Einführung & Motivation
- 2 Werkzeuge
 - eine unbekannte Datei, was nun?
 - selbstgestrickte Hilfsmittel
- 3 Containerformate
 - Textformate
 - Binärformate
- 4 Datenformate
 - Was braucht's für ein Modell?
 - Datentypen
- 5 Implementierung
- 6 Links

- 3D ist überall...
- ... trotzdem gibt es keinen universellen Dateibetrachter
- 3D-Formate sind nicht wirklich alle gut dokumentiert
- man kann Hexdumps ausdrucken und als Rätselblock verwenden

eine unbekannte Datei, was nun?

- file, immer gut (wenn man dran glaubt)
- strings, manchmal hilfreich
- xxd, Hauptsache hexadezimal
- \$EDITOR nach Wahl

selbstgestrickte Hilfsmittel

- ziemlich früh ein Prototyp-Plugin für die `libg3d`
- `g3d-stat`, quasi der Kommandozeilen-Client für die Bibliothek
- `detree`, irgendwelche Bäume angucken
- ein paar winzige Programme, um Hexadezimal- und Fließkommazahlen ineinander umzuwandeln

Textformate

- meist gut lesbar und recht einfach zu verstehen ...
- ... aber umständlich und aufwändig zu implementieren
- hierarchische oder sequentielle Formate
- neuerdings häufig XML-basiert
- einige 3D-Formate beschreiben Text- und Binärformat
- Textformate werden im Folgenden größtenteils ignoriert

- durchaus etwas kryptischer, aber maschinell einfacher zu lesen
- Endianness spielt eine Rolle
- meist hierarchisch (Ausnahmen existieren)
- oft in Chunks organisiert, gekennzeichnet durch Identifier und Längeninformatoren, unbekannte Chunks können übersprungen werden
- teilweise generische Container wie Zip, IFF oder Structured File (*aaarghhh*)

Was braucht's für ein Modell?

- Vertizes – Punkte im Raum
- Flächen – Dreiecke, Vie(r|l)ecke; meist durch Indizes der Eckpunkte gekennzeichnet
- Materialinformation (Farben, Texturen, Oberflächenbeschaffenheit)
- Normalen, Texturvertizes, Transformationsmatrizen ...
- Gruppierungen, Objektnamen ...

- '\0'-terminiert, feste Länge oder mit expliziter Längenangabe
- je nach Herkunft unterschiedliche Zeichenkodierung möglich, auch Multibyte-Strings nicht unüblich
- auch wichtig zur Identifizierung von Chunks innerhalb der Dateien

Integer-Typen

- meist 32 Bit, teils auch weniger oder mehr
- Indizes, Flags, Farbwerte, Abmessungen, Offsets, Dateigrößen. . .

Fließkommazahlen

- meist 32 oder 64 Bit nach IEEE 754 (z.B. 0x3f800000 für 1.0 – Big Endian)
- verwendet z.B. für Vertices und Matrizen

Blobs

- teilweise sind Texturen und andere Blobs direkt eingebettet
- einfach zu identifizieren, solange die Magic-Bytes dran sind

- Bibliothek mit vielen Hilfsfunktionen zur 3D-Basterei
- Plugins zum Laden von 3D-Modellen und Texturen
- unabhängig von 3D-APIs wie OpenGL
- verfügbar unter der LGPL

- Bildbetrachter-Äquivalent für 3D-Modelle
- relativ simpel
- Quellen unter der GPL

Links

- <http://automagically.de/g3dviewer/>
- <http://gna.org/projects/libg3d/>