



Aufgabe
Phase2

Thomas Behr

Motivation

Typen

Import

Export

Erzeugen

Transformieren

Verbinden

Zerlegen

Sonstige

Visualisierung

Aufgabenvorstellung Phase 2

3D-Algebra

Thomas Behr

Fakultät für Mathematik und Informatik
Datenbanksysteme für neue Anwendungen



FernUniversität in Hagen

20.November 2014

©2014 FernUniversität in Hagen



Das Ziel

Aufgabe
Phase2

Thomas Behr

Motivation

Typen

Import

Export

Erzeugen

Transformieren

Verbinden

Zerlegen

Sonstige

Visualisierung



- Repräsentation von Punkten, Flächen und Körpern im \mathbb{R}^3
- Import/Export von Standardformaten
- Objekterzeugung
- Manipulation einzelner Objekte (Drehen, Verschieben usw.)
- Mengenoperationen
- Anzeige der Objekte in der Javagui



Wozu und warum jetzt?

Aufgabe
Phase2

Thomas Behr

Motivation

Typen

Import

Export

Erzeugen

Transformieren

Verbinden

Zerlegen

Sonstige

Visualisierung

- Darstellung von Objekten der realen Welt
- 3D-Geräte für Heimgebrauch erschwinglich
 - 3D-Scanner
 - 3D-Drucker
- größere Datenmengen verfügbar
- Verwaltung in einer Datenbank sinnvoll
- SECONDO bietet noch keine Möglichkeit, 3D-Objekte zu speichern



3D-Daten – Woher kommen die?

Aufgabe
Phase2

Thomas Behr

Motivation

Typen

Import

Export

Erzeugen

Transformieren

Verbinden

Zerlegen

Sonstige

Visualisierung

- in Programmen konstruiert
- 3D-Scanner
 - Laser-Scanner, z.B. Makerbot Digitizer
 - Strukturiertes Licht, z.B. David SLS-2
 - Infrarotraster, z.B. Sense3D, Kinect
 - Ultraschallscanner
- Photogrammetrie



`point3d` Element aus dem \mathbb{R}^3

`surface3d` Oberfläche (Menge von Dreiecken)

`volume3d` Körper (Menge von Dreiecken)

weitere Typen

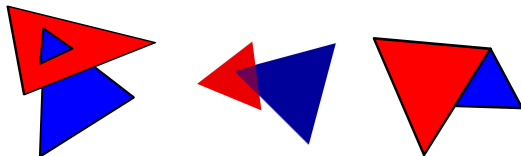
`vector3d` Richtungsvektor

`plane3d` Ebene



- dargestellt als Menge von Dreiecken
- müssen nicht zusammenhängend sein
- überlappungsfrei
- Dreiecke haben paarweise entweder
 - einen gemeinsamen Eckpunkt
 - eine gemeinsame Kante, d.h. zwei gemeinsame Eckpunkte
 - gar keine gemeinsamen Punkte

Nicht erlaubt:





- ebenfalls als Dreiecksmenge dargestellt
- Bedingungen für Oberflächen gelten auch hier
- zusätzlich:
 - Die Oberflächen sind wasserdicht, d.h. jede Kante eines Dreiecks ist auch Kante eines anderen Dreiecks.
 - Im Inneren eines Körpers darf kein weiterer Körper sein.
 - Jedem Dreieck ist ansehbar, auf welcher Seite das Innere des dargestellten Körpers liegt.



Import von Körpern

Aufgabe
Phase2

Thomas Behr

Motivation

Typen

Import

Export

Erzeugen

Transformieren

Verbinden

Zerlegen

Sonstige

Visualisierung

- STL-Format (Surface Tessellation Language)
- binär oder ASCII
- stellt Mengen von Dreiecken dar
- Datenquelle
`www.thingiverse.com`



STL-ASCII-Beispiel

Aufgabe
Phase2

Thomas Behr

Motivation

Typen

Import

Export

Erzeugen

Transformieren

Verbinden

Zerlegen

Sonstige

Visualisierung

```
solid liberty
  facet
    normal 2.03e-01 -8.889e-01 -4.10e-01
  outer loop
    vertex 3.94e+01 -7.22e+00 1.24e+02
    vertex 3.91e+01 -7.68e+00 1.25e+02
    vertex 3.87e+01 -7.45e+00 1.25e+02
  endloop
endfacet
...
endsolid
```



STL-BINARY-AUFBAU

Aufgabe
Phase2

Thomas Behr

Motivation

Typen

Import

Export

Erzeugen

Transformieren

Verbinden

Zerlegen

Sonstige

Visualisierung

Bytes Inhalt

0-79 Header, Inhalt nicht festgelegt

80-83 vorzeichenloser Integer: Anzahl der Dreiecke

84-... Dreiecke

Jedes Dreieck ist wie folgt aufgebaut:

Bytes

0-3, 4-7, 8-11

12-15, 16-19, 20 -23

24-27, 28-31, 32-35

36-39, 40-43, 44-47

48-49

50-.. ?

Inhalt

(n_1, n_1, n_2) : Normalenvektor

(x_1, y_1, z_1) : erster Punkt

(x_2, y_2, z_2) : zweiter Punkt

(x_3, y_3, z_3) : dritter Punkt

vorzeichenloser Integer: Byteanzahl eines am Dreieck hängenden Attributs, im Normalfall 0

optional: zusätzliches Attribut



Normalenvektor und Knotenorientierung

Aufgabe
Phase2

Thomas Behr

Motivation

Typen

Import

Export

Erzeugen

Transformieren

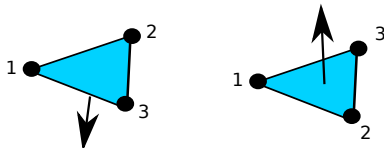
Verbinden

Zerlegen

Sonstige

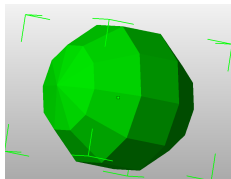
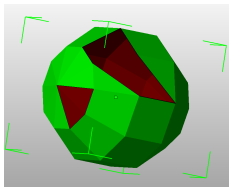
Visualisierung

- Normalenvektor
 - steht senkrecht auf dem Dreieck
 - zeigt nach außen
 - hat Länge 1
- Knoten sind entgegengesetzt zum Uhrzeigersinn orientiert, wenn man von außen auf das Dreieck sieht
- Außeninformation ist redundant gespeichert





- Viele STL-Dateien teilweise fehlerhaft
- Der **importSTL**-Operator korrigiert Fehler soweit möglich
- dies umfasst
 - Teilen sich schneidender oder überlappender Dreiecke
 - Entfernen doppelter Dreiecke
 - Schließen von Löchern
 - Umdrehen der Orientierung von Dreiecken
 - Entfernen von Dreiecken aus dem Objektinneren





- **exportSTL:** volume3d × text × string × bool → bool
Körper × Dateiname × Objektname × binär → Erfolg
exportiert einen Körper
- Prüfen der Korrektheit, z.B. mittels `net fabb`
Basisversion kostenlos



Erstellung einfacher Körper

Aufgabe
Phase2

Thomas Behr

Motivation

Typen

Import

Export

Erzeugen

Transformieren

Verbinden

Zerlegen

Sonstige

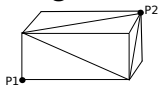
Visualisierung

Operator

Ergebnis

Argumente

createCube



point3d × *point3d*

createCylinder



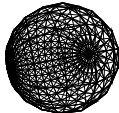
point3d × *real* × *real* × *int*

createCone



point3d × *real* × *real* × *int*

createSphere



point3d × *real* × *int*



Konvertierung von Regionen

Aufgabe
Phase2

Thomas Behr

Motivation

Typen

Import

Export

Erzeugen

Transformieren

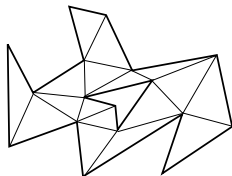
Verbinden

Zerlegen

Sonstige

Visualisierung

- **region2surface:** region \rightarrow surface3d
 - Triangulation einer Region
- **region2volume:** region \times real \rightarrow volume3d
 - Region mit einer Dicke
- **mregion2volume:** mregion \times real \rightarrow volume3d
 - bewegte Region
 - „Dicke“ eines Tages





Affine Transformationen

Aufgabe
Phase2

Thomas Behr

Motivation

Typen

Import

Export

Erzeugen

Transformieren

Verbinden

Zerlegen

Sonstige

Visualisierung

- Rotation, Skalierung, Verschiebung
- ändern die Lagebeziehungen von Objekten nicht
- lassen sich auf Matrixmultiplikationen zurückführen
- Operatoren sind nur im Aufbau der Matrix verschieden
- jeder Punkt des Objekts wird mit der Transformationsmatrix multipliziert
- Verwendung von 4-dim. Punkten und 4×4 Matrizen für \mathbb{R}^3

Beispielmatrix (Rotation um die x -Achse mit Winkel ϕ):

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi & 0 \\ 0 & \sin \phi & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



Operatoren

Aufgabe
Phase2

Thomas Behr

Motivation

Typen

Import

Export

Erzeugen

Transformieren

Verbinden

Zerlegen

Sonstige

Visualisierung

rotate rotiert Objekt um bestimmten Punkt und Achse mit geg. Winkel

mirror spiegelt Objekt an einer Ebene

translate verschiebt Objekt um einen Vektor

scaleDir skaliert ein Objekt aus einem Punkt in gegebene Richtung

scale gleichmäßige Skalierung aus einem geg. Punkt



Mengenoperationen

Aufgabe
Phase2

Thomas Behr

Motivation

Typen

Import

Export

Erzeugen

Transformieren

Verbinden

Zerlegen

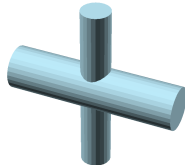
Sonstige

Visualisierung

union Vereinigung zweier Objekte

minus Differenz zweier Objekte

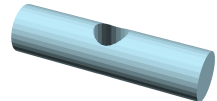
intersection Schnitt zweier Objekte



union



intersection



difference

zweier Zylinder



Zerlegen

Aufgabe
Phase2

Thomas Behr

Motivation

Typen

Import

Export

Erzeugen

Transformieren

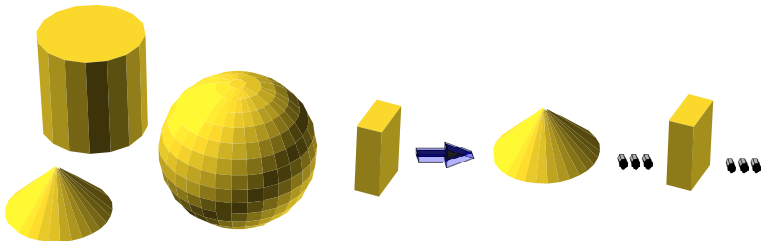
Verbinden

Zerlegen

Sonstige

Visualisierung

- Oberflächen und Körper können aus mehreren, nicht verbundenen Komponenten bestehen
- Operator **components** zerlegt Objekt in diese Komponenten
- Ergebnis ist ein Strom von Objekten





Sonstige Operatoren

Aufgabe
Phase2

Thomas Behr

Motivation

Typen

Import

Export

Erzeugen

Transformieren

Verbinden

Zerlegen

Sonstige

Visualisierung

size liefert die Anzahl der enthaltenen Dreiecke
bbbox liefert die Bounding Box eines Objekts als
rect3d



- eigener Viewer
- kann einzelne 3D-Objekte und Relationen darstellen
- Farbe für Objekte auswählbar
- Größe für Punkte auswählbar
- bei Relationen kann ausgewählt werden, ob alle Objekte in einer Farbe oder jedes Objekt in einer anderen Farbe dargestellt wird
- Standardfunktionalitäten (per Maus)
- Verwendung von java3d¹

¹Download unter

<https://java3d.java.net/binary-builds.html>



Standardfunktionalitäten

Aufgabe
Phase2

Thomas Behr

Motivation

Typen

Import

Export

Erzeugen

Transformieren

Verbinden

Zerlegen

Sonstige

Visualisierung

- Drehen
- Verschieben
- Zoom
- Setzen der Hintergrundfarbe
- Setzen/Entfernen von Lichtquellen
- Eigenschaften von Lichtquellen ändern



- Javagui bereits konfiguriert für Java3D-Verwendung
- Entpacken der Java3D-Bibliothek in beliebiges Verzeichnis
- Export zweier Umgebungsvariablen

```
J3D=$HOME/j3d/lib
J3DE=$J3D/ext
J3P=$J3DE/j3dcore.jar
J3D=$J3P:$J3DE/j3dutils.jar
J3P=$J3P:$J3DE/vecmath.jar
export JAVA3DPATH=$J3P
export JAVA3DLIB=$J3D/amd64
```

- z.B. in `.secondorc` Datei



Aufgabe
Phase2

Thomas Behr

Motivation

Typen

Import

Export

Erzeugen

Transformieren

Verbinden

Zerlegen

Sonstige

Visualisierung



Für Fragen bereit ich bin.