

Das Modellieren von 2D- und 3D-Objekten

Florian Buchegger, Michael Haberleitner

December 11, 2015

1 VTK-Format

Eine VTK-Datei besteht aus einem Header und einem Body. Während im Header lediglich wichtige Informationen enthalten sind, welche für das entschlüsseln der Datei notwendig sind, beinhaltet der Body die wesentlichen Daten der Geometrien.

1.1 Header

Die ersten drei Zeilen jeder Datei sind:

- Header # vtk DataFile Version 2.0
- Titel ein beliebiger Name (max 256 Zeichen)
- Datentyp ASCII oder BINARY

Beispiel:

```
#vtk DataFile Version 2.0
Meine coolen Daten
ASCII
```

1.2 Body

Danach kommen Blöcke von Daten, beginnend mit Codewörtern:

- DATASET *type* beschreibt den nun kommenden Geometrie-Typ

Es gibt verschiedene Geometrie-Typen. Wir verwenden nur UNSTRUCTURED_GRID.

- POINTS *n dataType* nun kommen *n* Zeilen mit Punkt-Koordinaten
- CELLS *m1 m2* nun kommen *m1* Zeilen mit Objekten und insgesamt *m2* Werten
- CELL_TYPES *m1* nun kommen *m1* Zeilen, welche die Objekt-Typen beschreiben

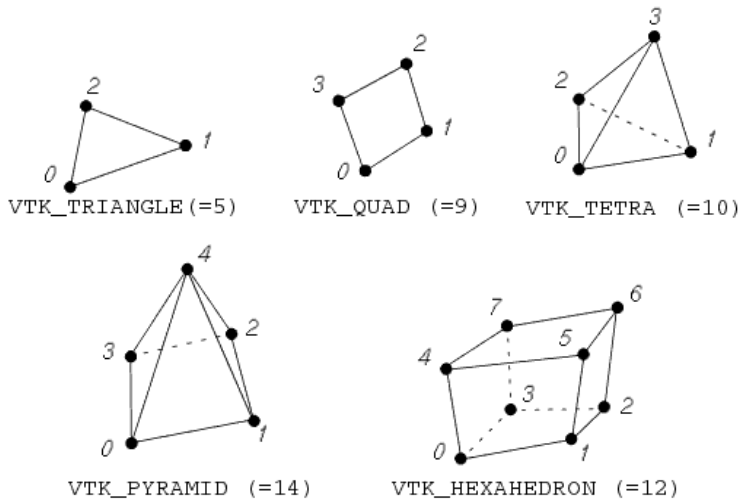


Figure 1: Die verschiedenen Typen von Objekten

Tipp:

Wenn Du die einzelnen Objekte in unterschiedlichen Farben darstellen möchtest, füge nach dem Block 'CELL_TYPES' noch einen weiteren Block ein. Dieser sieht aus wie folgt:

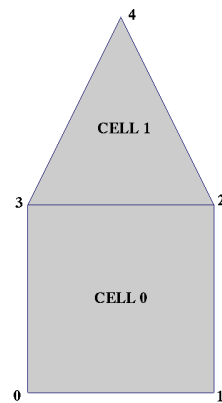
```
CELL_DATA m1
SCALARS cell_scalars int 1
LOOKUP_TABLE default
0
1
:
m1 - 1
```

1.3 Beispiel Haus

```
# vtk DataFile Version 2.0
Voll cooles Haus
ASCII
DATASET UNSTRUCTURED_GRID
POINTS 5 float
0 0 0
1 0 0
1 1 0
0 1 0
0.5 2 0

CELLS 2 9
4 0 1 2 3
3 2 3 4

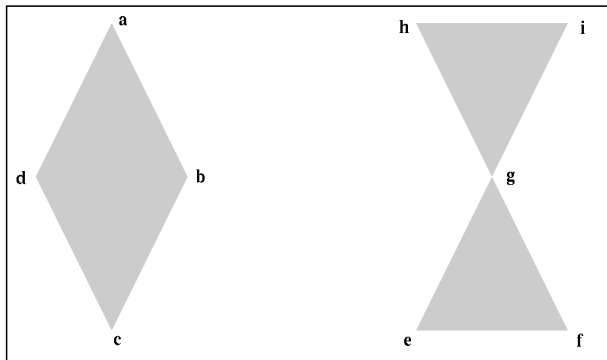
CELL_TYPES 2
9
5
```



1.4 Übungsbeispiele

Die folgenden Aufgaben sollen Euch dabei helfen, das Format besser zu verstehen. Versucht nun selber die folgenden Bilder zu rekonstruieren.

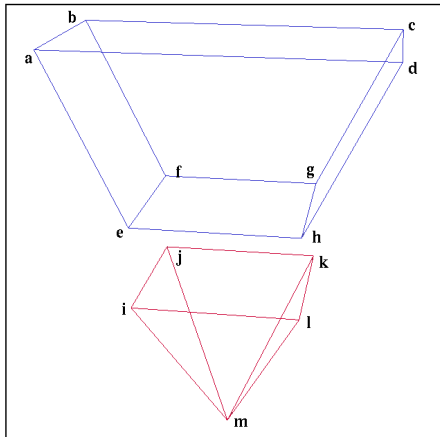
Beispiel 1: 2D-Objekte



$$\begin{aligned} a &= (0, 2) \\ b &= (0.5, 1) \\ c &= (0, 0) \\ d &= (-0.5, 1) \\ e &= (2, 0) \\ f &= (3, 0) \\ g &= (2.5, 1) \\ h &= (2, 2) \\ i &= (3, 2) \end{aligned}$$

Figure 2: 2D-Objekte

Beispiel 2: 3D-Objekte



$$\begin{aligned} a &= (-0.5, 0, 1) \\ b &= (-0.5, 1, 1) \\ c &= (1.5, 1, 1) \\ d &= (1.5, 0, 1) \\ e &= (0, 0, 0) \\ f &= (0, 1, 0) \\ g &= (1, 1, 0) \\ h &= (1, 0, 0) \\ i &= (0, 0, -0.5) \\ j &= (0, 1, -0.5) \\ k &= (1, 1, -0.5) \\ l &= (1, 0, -0.5) \\ m &= (0.5, 0.5, -1.5) \end{aligned}$$

Figure 3: 3D-Objekte

Beispiel 3: 2D-Sierpinski Dreieck

Das 2D-Sierpinski Dreieck ist ein Fraktal, also ein Objekt in welchem ein bestimmtes Muster immer wieder auftaucht. Man erhält es, indem man die Mittelpunkte der drei Kanten eines gleichseitigen Dreiecks miteinander verbindet, und somit drei kleinere Dreiecke erzeugt. Danach teilt man jedes dieser drei Dreiecke erneut in drei weitere Dreiecke auf und führt diese Konstruktion beliebig lange fort.

Konstruiere ein 2D-Sierpinski Dreieck Level 1.

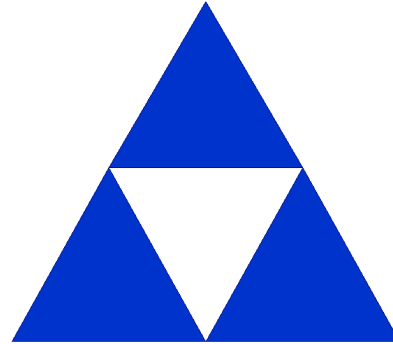


Figure 4: 2D-Sierpinski Level 1

Beispiel 4: 2D-Mengerschwamm

Der 2D-Mengerschwamm ist ebenfalls ein Fraktal. Man beginnt hier mit einem Quadrat und teilt dieses in 9 gleich große Quadrate auf, wobei das mittlere aus dem Objekt entfernt wird. Das neue Objekt besteht also aus 8 Quadraten, welche im nächsten Level wiederum in 8 kleinere Quadrate aufgeteilt werden.

Konstruiere einen 2D-Mengerschwamm Level 1.

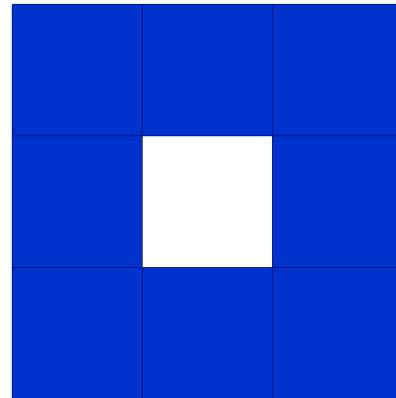


Figure 5: 2D-Sierpinski Level 1

2 Automatisierung mit Python

Python ist eine Programmiersprache, mit deren Hilfe wir größere Probleme lösen können.

2.1 Programmierumgebung

Die Programmierumgebung 'Idle' könnt ihr folgendermaßen aufrufen:

- Drück 'Alt+F2' um die Quickstartbar zu aktivieren.
- Schreibe 'idle' hinein.
- Drücke 'Enter'.

Folgende Tipps sind auch hilfreich:

- 'Strg+F6' führt einen restart der Shell durch.
- 'File' → 'New File' öffnet ein neues Skript.
- 'F5' in einem Skript führt es aus.
- Alles was in einem Skript nach '#' in der Zeile steht, wird nicht ausgeführt.

2.2 Übungsbeispiele

Folgende Aufgaben sollen mit Hilfe von Python gelöst werden:

Beispiel 1: Punkte auf Kreis:

Erzeuge auf dem Einheitskreis (Radius = 1 und Mittelpunkt = $\{0,0\}$) eine gegebene Anzahl an Punkten und gib sie in einer Liste am Bildschirm aus.

Beispiel 2: Reihe von Würfeln:

Erzeuge eine Funktion 'UnitCubes(n)', die eine gegebene Anzahl an Einheitswürfeln (Kantenlänge = 1) erzeugt, die entlang der x-Achse auf einer Linie liegen. Der erste Würfel liegt zwischen $\{0,0,0\}$ und $\{1,1,1\}$, der zweite Würfel zwischen $\{1,0,0\}$ und $\{2,1,1\}$, etc. Die Ausgabe sollte eine Liste sein, die alle Eckpunkte der Würfeln in der richtigen Reihenfolge beinhaltet (siehe VTK.HEXAHEDRON). Der Aufruf 'UnitCubes(5)' sollte 5 Würfel berechnen.

Beispiel 3: Sierpinski 3D:

Erzeuge eine Sierpinski Pyramide (Erklärung auf Tafel).

Beispiel 4: Menger-Schwamm 3D:

Erzeuge einen 3D-Menger Schwamm (Erklärung auf Tafel).

Beispiel 5: Weihnachtssterne:

Wer kann den schönsten, 3-dimensionalen Weihnachtsstern erzeugen?