



ProPra WS 2008/09: „LEGO-Sculptor“

Dipl.-Inform. Christian Düntgen

Worum geht es?

1. Einlesen einer Formbeschreibung aus VRML-Datei
(ggf. Visualisierung der Form)
2. Erzeugung einer Liste zu verwendender Bausteintypen
3. Einstellen von Parametern
4. Automatische Konstruktion eines Modells
Anzeige des Fortschritts während der Berechnung
5. Ausgabe des fertigen Bauplans als LDraw-Datei
Ausgabe einer Teileliste
(ggf. Visualisierung des Modells)



Der LDraw Editor MLCAD

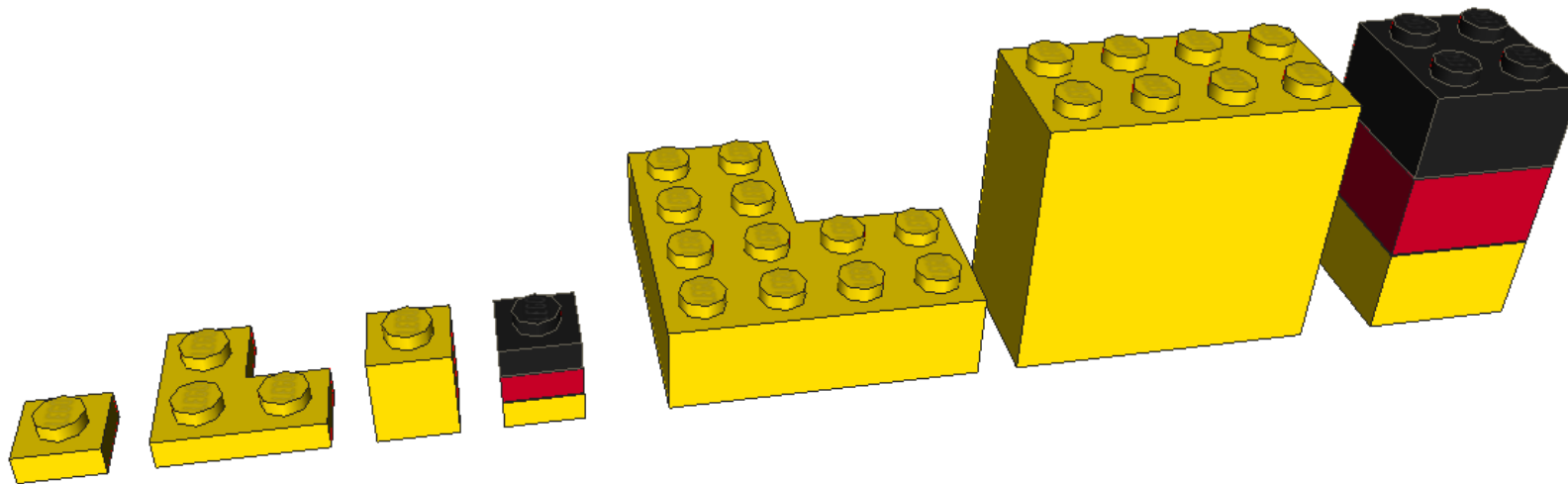
The screenshot shows the LDraw Editor MLCAD interface. On the left is a 'Brick' library with a grid of various colored and shaped bricks. The main workspace is divided into four views: Front, Left, Top, and 3D. The 3D view shows a pyramid constructed from blue, red, yellow, and black bricks. A table in the center lists the parts used in the model.

| Type | Color | Position | Rotation | Part no./Model... | Part name/Description |
|---------|-------|--------------------|--|-------------------|--|
| CDMM... | .. | | | | Pyramid |
| CDMM... | .. | | | | Name: PYRAMID.ldr |
| CDMM... | .. | | | | Author: James Jessiman |
| CDMM... | .. | | | | Original LDraw Model - LDraw beta 0.27 Archive |
| CDMM... | .. | | | | Pyramid |
| PART | Blue | -40.000,-24.000... | 1.000,0.000,0.000 0.000,1.000,0.000... | 3001.DAT | Brick 2 x 4 |
| PART | Blue | 40.000,-24.000... | 1.000,0.000,0.000 0.000,1.000,0.000... | 3001.DAT | Brick 2 x 4 |
| PART | Blue | 60.000,-24.000... | 0.000,0.000,1.000 0.000,1.000,0.000... | 3001.DAT | Brick 2 x 4 |
| PART | Blue | 40.000,-24.000... | 1.000,0.000,0.000 0.000,1.000,0.000... | 3001.DAT | Brick 2 x 4 |
| PART | Blue | -40.000,-24.000... | 1.000,0.000,0.000 0.000,1.000,0.000... | 3001.DAT | Brick 2 x 4 |
| PART | Blue | -60.000,-24.000... | 0.000,0.000,1.000 0.000,1.000,0.000... | 3001.DAT | Brick 2 x 4 |
| STEP | .. | | | | |
| PART | Red | -20.000,-48.000... | 1.000,0.000,0.000 0.000,1.000,0.000... | 3001.DAT | Brick 2 x 4 |
| PART | Red | 40.000,-48.000... | 0.000,0.000,1.000 0.000,1.000,0.000... | 3001.DAT | Brick 2 x 4 |
| PART | Red | 20.000,-48.000... | 1.000,0.000,0.000 0.000,1.000,0.000... | 3001.DAT | Brick 2 x 4 |
| PART | Black | 40.000,-48.000... | 0.000,0.000,1.000 0.000,1.000,0.000... | 3001.DAT | Brick 2 x 4 |

Annotations:

- Baustein-Bibliothek (Textbeschreibungen)**: Points to the left sidebar library.
- Bausteinbibliothek (graphische Darstellung)**: Points to the brick grid in the bottom-left.
- Teile-Platzierung ähnlich LDraw-Format (Klartext statt Codes)**: Points to the parts list table.
- Arbeitsfläche mit 3 Projektionen und perspektivischer Ansicht**: Points to the four view windows (Front, Left, Top, 3D).

Geometrie des LEGO-Steins (1)



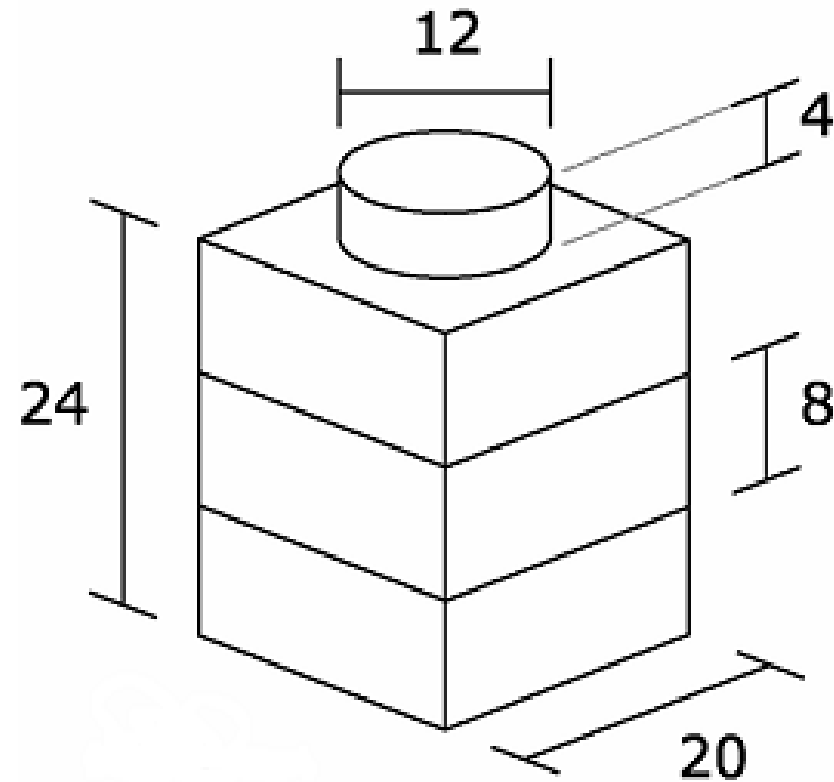
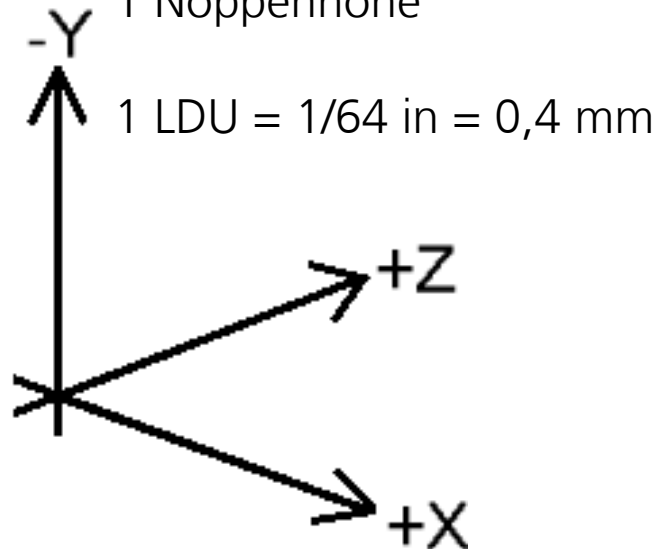
Platten (Plates): 1x1, 1x2, 1x3, 1x4, 1x6, 1x8, 1x10, 2x2, 2x2 Corner, 2x3, 2x4, 2x6, 2x8, 2x10, 2x12, 4x4, 4x4 Corner, 4x6, 4x8, 4x10, 4x12, 6x6, 6x8, 6x10, 6x12, 6x14, 6x16, 6x24, 8x8, 8x11.

Steine (Bricks): 1x1, 1x2, 1x3, 1x4, 1x6, 1x8, 1x10, 1x12, 1x16, 1x1x5, 1x2x2, 1x2x5, 1x3x5, 1x6x5, 2x2, 2x2 Corner, 2x3, 2x4, 2x6, 2x8, 2x10, 2x2x3, 2x4x3, 2x6x3, 4x4 Corner, 4x6, 4x10, 4x12, 4x18, 8x8, 8x16.

Geometrie des LEGO-Steins (2)

Bausteine werden in LDraw Units (LDU) bemaßt:

| | |
|----------------------|----------|
| 1 Steinbreite/-tiefe | = 20 LDU |
| 1 Steinhöhe | = 24 LDU |
| 1 Plattenhöhe | = 8 LDU |
| 1 Noppendurchmesser | = 12 LDU |
| 1 Noppenhöhe | = 4 LDU |



VRML

- Hierarchischer Szenen-Aufbau (**Baumstruktur**) in einer Textdatei
- **Knoten**: [DEF <name>] <nodeType> { <body> } | USE <name>
- **Felder**: <fieldName> <fieldValue> | <fieldName> [<fieldValues>]
- Verwendung von Makros (**USE**-Knoten: Unterstützung optional)
- **Erkennen irrelevanter Deklarationen**: Klammerungstiefe nachhalten

Einlesen der VRML-Datei

Auszuwertende Knoten:

- **Viewpoint:** Feld "position"
- **Shape:** Feld "geometry" mit den Klassen (Attributen)
 - Box (size),
 - Sphere (radius),
 - Cylinder (height, radius),
 - Cone (bottomRadius, height).
- **Group:** nur Feld "children" (mit Knoten-Liste)
- **Transform:** Felder center, children (mit Knoten-Liste), rotation, scale, scaleOrientation, translation.

Insbesondere **nicht** auszuwertende Knoten:

- **IS-, ROUTE-, PROTO- und EXTERNPROTO-Statements**

```
#VRML V2.0 utf8
Transform {
  rotation 0.124534 0.658936 -0.741819 0.120680
  translation 0.732158 0 4.720851
  children [
    Shape {
      geometry Box { size 2 3 1 }
      appearance Appearance {
        material Material { diffuseColor 1 1 1 }
        texture ImageTexture { url "marble2.gif" }
      }
    }
    Shape {
      geometry Box { size 1 3 1 }
      appearance Appearance {
        material Material { diffuseColor 1 1 1 }
      }
    }
  ]
  NavigationInfo { type "EXAMINE" }
  Background { skyColor 1 1 1 }
}
```

Irrelevante Teile
sind **hellgrau**
dargestellt

Konstruktion des Modells

Möglicher Ansatz:

1. Bestimme Dimensionen des Modells in LDU
2. Verwende Zellen (Voxel) der Größe $20 \times 20 \times 8$ LDU (= Plate 1x1)
3. Für jede Zelle: Bestimme, ob gefüllt oder leer
 - **Platziere größere Steine direkt, wann immer möglich**
 - **Ersetze später benachbarte Zellen durch größere Steine**
4. Stabilität:
 - **Baue Modell von unten nach oben auf.**
 - **Platziere Steine möglichst überlappend.**
 - **„Platten sind Freunde!“**

Immer die optimale Lösung finden zu wollen ist utopisch! Daher:
Heuristiken finden, die meistens ziemlich gut bauen.

Erzeugen der LDraw-Datei

- Dateiformat: Textdatei mit Endung „.ldr“
- Jede Zeile beschreibt die Platzierung eines Steins:
- **1 1 -40 -24 60 1 0 0 0 1 0 0 0 1 3001.dat**
- Bedeutung:
 - <linetype>:** 1 = einzubindende Datei
 - <color>:** 1 = schwarz
 - <position>:** x y z
 - <transformation matrix>:** a b c d e f g h i
 - <file>:** Brick 2x4

$$\begin{pmatrix} a & d & g & 0 \\ b & e & h & 0 \\ c & f & i & 0 \\ x & y & z & 1 \end{pmatrix}$$

Affine Transformationsmatrix

Transformation eines Punktes $x = (u, v, w)$

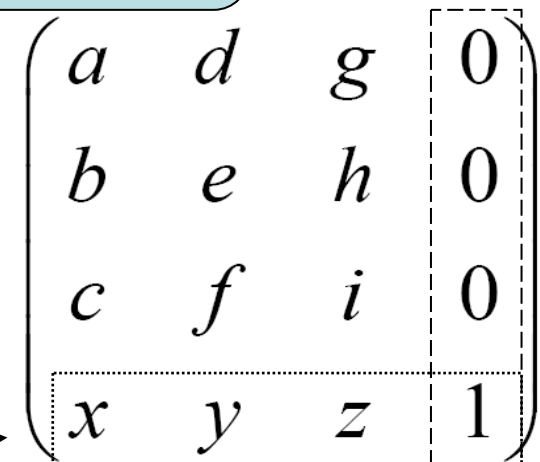
- Rotation
- Translation
- Skalierung

Multiplikation des Vektors x mit der Matrix A liefert Ergebnisvektor $x' = (u', v', w')$:

$$(u', v', w', 1)^T = A(u, v, w, 1)$$

- Umkehrfunktion zu A : A^{-1}
 A^{-1} ist die zu A inverse Matrix

Extraspalte und -zeile erlauben Translation und Rotationen in einem Durchgang, wenn man an x eine 1 anhängt



$$u' = au + bv + cw + x$$

$$v' = du + ev + fw + y$$

$$w' = gu + hv + iw + z$$

Affine Transformationsmatrizen (2)

Translation: $\underline{T} = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$ $\underline{T} \cdot (x, y, z, 1)^T = (x + t_x, y + t_y, z + t_z, 1)$

Rotation um x-Achse: $\underline{R}_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

Rotation um y-Achse: $\underline{R}_y = \begin{pmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

Rotation um z-Achse: $\underline{R}_z = \begin{pmatrix} \cos \gamma & -\sin \gamma & 0 & 0 \\ \sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

Skalierung: $\underline{S} = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ $\underline{S} \cdot (x, y, z, 1)^T = (s_x \cdot x, s_y \cdot y, s_z \cdot z, 1)^T$

~~Perspektive: $\underline{P} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{pmatrix}$ $\underline{P} \cdot (x, y, z, 1)^T = (x, y, z, \frac{z}{d})^T$~~

~~Orthogonale Projektion: $\underline{P}_{orth, z=0} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ $\underline{P}_{orth, z=0} \cdot (x, y, z, 1)^T = (x, y, 0, 1)^T$~~

Achtung:
Im Vergleich zur letzten Folie
wurden die Matrizen hier
transponiert!

Projektion und
Perspektive sollten
nicht erforderlich sein!

Weitere Quellen

- <http://de.wikipedia.org/wiki/Koordinatentransformation>
- [http://de.wikipedia.org/wiki/Homogene Koordinaten](http://de.wikipedia.org/wiki/Homogene_Koordinaten)
- LDraw: <http://www.ldraw.org/>
- LEGO Steine und Modelle: <http://www.peeron.com>
- Community: <http://www.lugnet.com/>