

LEGO-Modellierwerkzeuge

Einsatzmöglichkeiten des virtuellen Bauens in wissenschaftlichen Bereichen

Christian Ofenböck

Zusammenfassung—Nach Jahren als führendes Produkt im Spielzeugsektor wurde Lego[®] in den letzten Jahren auch als Mittel für wissenschaftliche Untersuchungen entdeckt. Neben Bereichen wie Verhaltens- und Lernforschung spielt das Baustein-konzept heute auch in der Computergraphik eine immer größer werdende Rolle. So sind etwa Studien zu Benutzerverhalten und die Suche möglicher Eingabesysteme für virtuelle Umgebungen mit Hilfe von Lego[®]-basierenden Systemen einfacher als mit klassischen Methoden zu gestalten. Mittlerweile existiert eine beachtliche Zahl an Modellierwerkzeugen, die in dieser Arbeit grob sondiert werden sollen. Ziel ist es, herauszufinden, ob diese in den oben genannten Bereichen sinnvoll eingesetzt werden könnten.

Schlüsselwörter—Lego[®], CAD, LDraw, Lego[®] Digital Designer, Modellierung

I. EINLEITUNG

Lego[®] ist eine eingetragene Marke der LEGO-Gruppe, die die vorliegende Arbeit weder autorisiert noch fördert.

Die vorliegende Arbeit verfolgt das Ziel, eine Evaluierung von Lego-Modellierwerkzeugen vorzunehmen, um zu zeigen, welche von ihnen für den Einsatz im wissenschaftlichen Bereich geeignet sind. Dazu wird in Kapitel II geklärt, warum Lego „mehr als nur Spielzeug ist“. Im Anschluss wird das LDraw-Projekt (Kapitel III) und in Kapitel IV eine Auswahl an LDraw-basierten Programmen betrachtet. Das offizielle Werkzeug Lego Digital Designer (Kapitel V) sowie eine kleine Auswahl weiterer Modelliertools (Kapitel VI) werden gefolgt von einer überblicksmäßigen Sichtung, wie die behandelten Werkzeuge wissenschaftlich relevant sein könnten (Kapitel VII).

II. LEGO - MEHR ALS NUR EIN SPIELZEUG

Bereits seit mehr als fünfzig Jahren ist Lego ein fester Bestandteil unserer Gesellschaft. Doch erst sehr spät wurde erkannt, dass es sich hierbei nicht nur um das bestverkaufte Spielzeug aller Zeiten handelt. Lego kombiniert Einfachheit und Komplexität wie kaum ein anderes verfügbares Produkt und ist daher auch für Forschungszwecke sehr wertvoll. In den Bereichen Verhaltens-, Unterrichts- und Lernprozessforschung konnten mit Hilfe der kleinen Bausteine viele neue Erkenntnisse gewonnen werden. Dazu hat vor allem die „Lifelong Kindergarten Research Group“ am MIT Media Lab beigetragen ([1], [2], [3]). In Kooperation mit dem MIT wurde auch die Lego-Mindstorms-Serie [4] entwickelt. Hierbei handelt es sich um eine Erweiterung des herkömmlichen Sets an Steinen um programmierbare Bauteile, Sensoren und Motoren, mit denen interaktive Systeme gebaut werden können.

Auch in der Informationsverarbeitung kann man aus der Verwendung des Spielzeuges wichtige Erkenntnisse ziehen.

Vor allem für die Planung und Realisierung möglichst intuitiver Eingabesteuerung in CAD-Anwendungen und virtuellen Umgebungen erweisen sich Studien mit Lego als sehr wertvoll. Baradaran [5] zeigte, dass auch Personen, die noch wenig Erfahrung im Umgang mit CAD-Anwendungen hatten, sich in einer dem Lego-Prinzip nachempfundenen Modellier-Umgebung schnell zurechtfinden, wenn auch die Bearbeitungszeit zum Erfüllen einfacher Aufgaben um ein Vielfaches höher war als mit realen Steinen. Dazu wurde das von Oh *et al.* [6] entwickelte System Virtual Lego (siehe Kapitel VI-B) verwendet.

III. LDRAW

LDraw ist ein System zum Abbilden von realen Legosteinen und dem Erstellen daraus zusammengesetzter Modelle. Das System wurde ursprünglich 1995 [7] von James Jessiman entwickelt und bestand damals aus dem DOS-basierten Editor LEdit, dem Betrachtungswerkzeug Ldraw und der LDraw-Dateiformat-Spezifikation. Das Dateiformat LDR (früher auch: DAT) wurde sehr früh zu einem Standard für Lego-basierte CAD-Anwendungen. Innerhalb weniger Jahre entstanden rund um LDraw viele weitere Werkzeuge, die allesamt auf diesem Format basierten [7]. Da Jessiman 1997 starb, wurden LEdit und Ldraw nicht weiter aktualisiert, blieben aber im Paketumfang enthalten.

A. Das LDraw-Dateiformat LDR

Das von Jessiman entwickelte Format wird nach wie vor - allerdings erweitert um einige Zusätze - verwendet. Das LDraw-Format ist textbasiert und daher auch ohne graphische Modellierwerkzeuge gut umsetzbar.

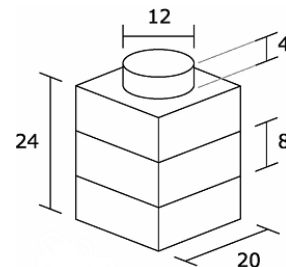


Abb. 1. Standard-1x1-Lego-Stein. Maßangaben in LDraw Units. [8]

Nach der Spezifikation [8] arbeitet es mit einem rechtshändigen Koordinatensystem mit eigenen Maßeinheiten, den LDraw Units (LDU). Ein LDU entspricht in etwa 0.4 Millimetern oder der Höhe eines Lego-Platten-Bausteins.

Demnach besitzt ein standardmäßiger 1x1-Lego-Baustein (Abbildung 1) eine Länge und Breite von je 20 LDU, eine Höhe von 24 LDU und einen Noppen, der weitere 4 LDU nach oben ragt.

Das LDR-Format besticht durch seinen simplen Aufbau: Jeder Zeile wird mittels einem vorangestellten numerischen Indikator ein Typ zugewiesen, der anzeigt, welcher Befehl folgt. Einzig eine vorangestellte Null hat zwei Bedeutungen und kann entweder von einem Kommentar oder einem Meta-Kommando gefolgt werden. Die Unterscheidung basiert hier auf definierten Schlüsselwörtern, die solche Meta-Befehle einleiten. Im Standard finden sich aus historischen Gründen einige wenige dieser Schlüsselwörter, obwohl im Jahr 2003 Dateiformat-unspezifische Definitionen davon ausgeschlossen und in eigenen Spracherweiterungen definiert wurden.

1) *Syntax*: Die Befehlssyntax gestaltet sich wie folgt [8]:

- 0: Kommentar, Meta-Befehl. `0 <Kommentar>, 0 CLEAR`
- 1: Datei-Verweis. `1 <colour> x y z a b c d e f g h i <file>`

Die so indizierten Zeilen bilden Verweise auf andere LDR-Dateien, die an dieser Stelle eingebunden werden. Dabei werden die zu verwendende Standardfarbe, ein Koordinatenpunkt (x, y, z) und eine 3x3-Matrix (a - i), sowie die zu ladende Datei angegeben. Die zu übergebende Matrix ist der obere linke Teil einer 4x4-Transformationsmatrix.

- 2: Linie. `2 <colour> x1 y1 z1 x2 y2 z2`

Es wird eine Verbindungslinie zwischen den beiden gegebenen Punkten in der angegebenen Farbe hergestellt. Linien werden vor allem zur Kantenvisualisierung verwendet.

- 3: Dreieck. `3 <colour> x1 y1 z1 x2 y2 z2 x3 y3 z3`

Die drei gegebenen Punkte bilden die Eckpunkte für ein in der angegebenen Farbe gefülltes Dreieck.

- 4: Viereck. `4 <colour> x1 y1 z1 x2 y2 z2 x3 y3 z3 x4 y4 z4`

Diese Anweisung zeichnet ein mit der angegebenen Farbe gefülltes Viereck. Die Eckpunkte können im oder gegen den Uhrzeigersinn angegeben werden und müssen ein konvexes Polygon einschließen.

- 5: Optionale Linie. `5 <colour> x1 y1 z1 x2 y2 z2 x3 y3 z3 x4 y4 z4`

Wenn eine durch die beiden Kontrollpunkte x3 und x4 gelegte Gerade die zu einer Geraden verlängerte Linie zwischen x1 und x2 nicht schneidet, wird zwischen x1 und x2 eine Linie in der angegebenen Farbe gezeichnet.

2) *Meta-Befehle*: Die in der Spezifikation vorhandenen Meta-Befehle werden als offiziell angesehen und müssen von jedem LDraw-Werkzeug behandelt werden. Von anderen Entwicklern oder Programmen definierte Befehle sind in einer Liste inoffizieller Meta-Befehle gesammelt. Die wichtigsten offiziellen Kommandos lauten:

- 0 PAUSE
Dieser Befehl verursacht ein Halten der Zeilen-Verarbeitung, bis eine Tastatur-Eingabe erfolgt.
- 0 SAVE
An dieser Stelle wird ein Bitmap des aktuell erstellten Modells im LDraw-Installationsverzeichnis angelegt.

- 0 STEP
Die Step-Anweisung ist eine Kombination aus Pause und Save und wird sehr häufig verwendet, um einzelne Bauschritte klar voneinander zu trennen. Sie spielt vor allem dann eine große Rolle, wenn Bauanleitungen in Einzelbildern erstellt werden sollen.
- 0 CLEAR
Clear löscht alle bis zu seiner Verarbeitung gezeichneten Teilmodelle, Teile und Primitive.
- Header-Informationen.
In den vorgeschriebenen Header-Informationen sind Angaben über Ersteller, Modell- und Dateibezeichnung zu machen.
- Typ-Informationen.
Typ-Informationen werden in Kapitel III-A.4 behandelt.

3) *Farben*: Für Farbangaben existiert eine eigene Referenztabelle (Abbildung 2), in der jeder Farbe eine eindeutige Nummer zugewiesen wird. Dabei gibt es zwei Sonderbelegungen:

- 16: Hauptfarbe.
Bei dieser Angabe wird für das zu zeichnende Objekt die vorgegebene Farbe verwendet. Sinnvoll eingesetzt kann diese Belegung also dann werden, wenn das Objekt mittels einer Verweis-Anweisung eines anderen Objektes gezeichnet wird. Der Farbparameter dient in diesem Fall als Hauptfarbe.
- 24: Komplementärfarbe.
Jeder in LDraw vorhandenen Farbe wird eine Komplementärfarbe zugewiesen. Jene der aktuellen Hauptfarbe wird meist für Ränder verwendet.

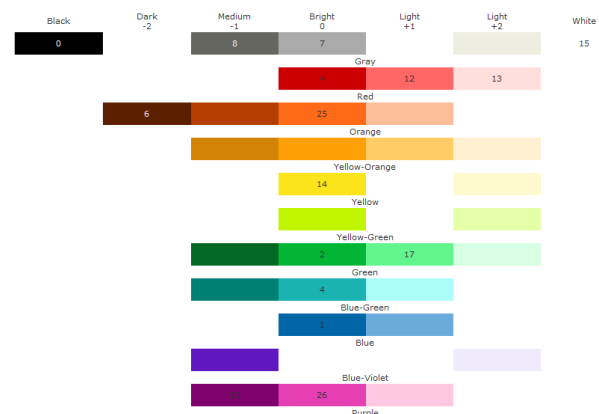


Abb. 2. Lego-Farben und ihre dazugehörigen Farbreferenzen in LDraw. 0 steht für Schwarz, 15 für Weiß. Leere Felder zeigen an, dass diese Farbe in Lego nicht existiert, sehr wohl jedoch in LDraw. [9].

4) *Dateitypen*: Da in LDR-Dateien unterschiedliche Typen gespeichert werden, steht in jeder Datei eine Meta-Anweisungszeile, die den Dateityp kennzeichnet. Dateien aus der LDraw.org-Bibliothek verwenden dazu das Schlüsselwort `!LDR_DRAW_ORG`, gefolgt von einem Typ-Parameter. Mögliche Ausprägungen dieses Parameters sind etwa `primitive` (Linie, Dreieck, Viereck), `part` (Baustein) oder `model` (aus mehreren Bausteinen gefertigtes Teil oder Modell) [8]. Dateien aus anderen Bibliotheken werden stattdessen mit dem

Schlüsselwort `Unofficial` versehen. Die Endung `DAT` wird mittlerweile nur noch für Dateien der LDraw.org-Bauteil-Bibliothek verwendet, alle anderen Dateien sollten mit der Endung `LDR` oder `MPD` versehen werden.

5) *Erweiterungen:* Einige Erweiterungen, die in LDraw eingebaut wurden, sind auch in die Spezifikation aufgenommen worden:

- **MultiPartDat (MPD).**
Diese Erweiterung erlaubt es, mehrere LDraw-Dateien in einer Datei zusammenzufügen. Mit Hilfe des Meta-Befehls `0 FILE <file>` wird der Beginn einer Datei gekennzeichnet, wobei ein entsprechender Dateiname anzugeben ist [10].
- **CATEGORY und KEYWORDS.**
Hiermit wurden weitere Meta-Befehle eingeführt, die zum Vermerken von Schlüsselwörtern und Kategorien in LDraw-Dateien verwendet werden. Das erleichtert die Einordnung einzelner Dateien in die entsprechenden Abschnitte der LDraw.org-Bibliothek. Dazu wurden alle vorhandenen Teile in etwa 70 thematische Kategorien aufgespalten, die das Referenzieren zusammengehöriger Teile erleichtern [11].

```

1 0 Brick 1 x 1
2 0 Name: 3005.dat
3 0 Author: James Jessiman
4 0 Original LDraw Part
5 0 LDRAW_ORG Part UPDATE 2002-03
6
7 0 BFC CERTIFY CCM
8
9 0 2002-05-07 KJM BFC Certification
10
11 0 BFC INVERTNEXT
12 1 16 0 24 0 6 0 0 0 -20 0 0 0 6 box5.dat
13
14 4 16 10 24 10 6 24 6 -6 24 6 -10 24 10
15 4 16 -10 24 10 -6 24 6 -6 24 -6 -10 24 -10
16 4 16 -10 24 -10 -6 24 -6 6 24 -6 10 24 -10
17 4 16 10 24 -10 6 24 -6 6 24 6 10 24 10
18
19 1 16 0 24 0 10 0 0 0 -24 0 0 0 10 box5.dat
20
21 1 16 0 0 0 1 0 0 0 1 0 0 0 1 stud.dat
22 0

```

Abb. 3. Definition eines Standard-1x1-Bausteins im LDR-Format mit Koordinatenangabe in LDU. Die Farbe kann bei Verwendung über Parameter gesteuert werden.

B. Modellieren mit LDraw

LEdit und Ldraw sind im Umfang des Standard-LDraw-Pakets zwar enthalten, finden aber kaum noch Anwendung. Alternative Programme werden im folgenden Kapitel IV betrachtet. Das Erstellen von LDraw-Modellen basiert auf dem zuvor (Kapitel III-A) beschriebenen Standard. Mit dessen Hilfe werden sowohl kleinste Bausteine als auch ganze Szenen erstellt, indem die Einzelteile über Verweis-Zeilen zusammengefügt werden. Mittels Meta-Befehlen können auch Bauschritte markiert werden.

IV. LDRAW-BASIERTE WERKZEUGE

LDR-Daten bestehen aus lesbaren Anweisungen und können daher mit Hilfe eines einfachen Texteditors erzeugt werden. Eine komfortablere Lösung bieten allerdings die im Folgenden genannten Werkzeuge, da sie graphische Modellier-Oberflächen zur Verfügung stellen und die entsprechenden LDR-Anweisungen generieren.

A. MLCad

MLCad [12] wurde von Michael Lachmann entwickelt. Die letzte Aktualisierung erfolgte 2005 mit der Herausgabe der Version 3.20, für die auch eine deutschsprachige Sprachdatei vorhanden ist. MLCad ist kein völlig eigenständiges Programm, sondern lediglich ein Werkzeug, mit dem LDraw-Dateien erzeugt, gelesen und bearbeitet werden können. Im Gegensatz zu den LDraw-Standardprogrammen verfügt MLCad über eine graphische Benutzeroberfläche. Modelliert wird in mehreren 2D-Ansichten, gleichzeitig wird die aktuelle Szene als 3D-Modell visualisiert.

1) *Platzieren, bewegen und rotieren:* Während das Bewegen von Teilen mit Drag & Drop sich in MLCad als sehr vorteilhaft erweist, werden Rotationen ausschließlich über die Angabe von Rotationspunkten und -winkeln gelöst. Alle benötigten Maßangaben haben in LDU zu erfolgen.

Da MLCad Bausteine als stereometrische Objekte behandelt, fehlt das aus Lego bekannte Andocken („Snapping“) [13] und Steine können auch versetzt (nicht deckend an den Noppen) aneinander gereiht werden (Abbildung 4). Verändert man die Position eines Steines, gibt es keine Restriktionen. Teile können so auch „ineinander“ platziert werden, was eine grobe Abweichung von der „Lego-Realität“ darstellt (Abbildung 5).

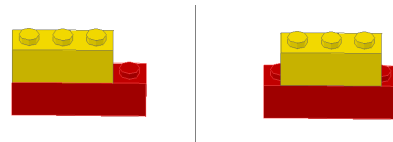


Abb. 4. links: Bausteine deckend an den Noppen ausgerichtet; rechts: keine Deckung, der gelbe Baustein ist um 5 LDU verschoben.

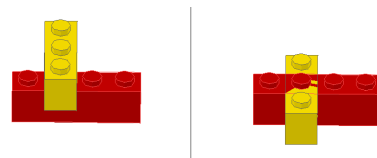


Abb. 5. links: richtig angeordnete Bausteine; rechts: „ineinander“ verschobene Bausteine.

Die Objekte werden an einem in seiner Feinheit einstellbaren Raster (Breite: 10, 5 oder 1 LDU) ausgerichtet. Das erschwert zwar für Einsteiger das Modellieren (im Vergleich zu einer Snapping-Funktion), ist aber für erweiterte Anwendungen enorm hilfreich. Einzelne Bausteine können gruppiert und gemeinsam bearbeitet werden.

2) *Leistungsmerkmale:* MLCad ist ein sehr leistungsfähiges Werkzeug, das mit einer Vielzahl an Aktionsmöglichkeiten aufwartet. Alle im LDR-Standard definierten Befehle können eingefügt werden, was bedeutet, dass neben Bausteinen und Objekten auch Primitive (Linien, Dreiecke, Vierecke) erzeugt werden können. Darüberhinaus bieten Modellgeneratoren die Möglichkeit, Minifiguren und Rotationsobjekte einzufügen oder aus Bitmaps LDraw-Mosaik zu erzeugen. Bei zu großen Modellen kommt es allerdings durch hohe Speicherlastigkeit zu einer starken Leistungseinbuße. Abhilfe schafft das Wegschalten nicht benötigter Ansichten und das Ausblenden der Noppen.

MLCad ist sowohl auf Windows- als auch Linux-Systemen lauffähig und in acht Sprachversionen erhältlich.

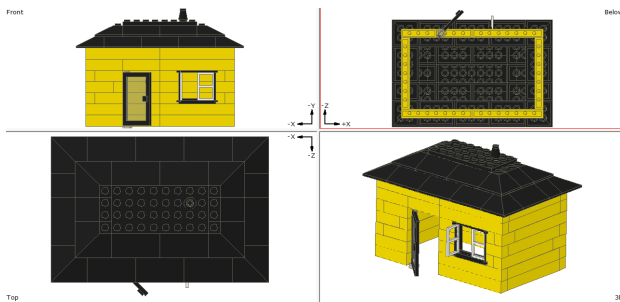


Abb. 6. Mit MLCad erstelltes Modell eines Hauses bestehend aus 81 Bauteilen in den drei Modellieransichten und als 3D-Modell.

B. LeoCAD

Ein weiteres graphisches Modellierwerkzeug für LDraw-Modelle stellt LeoCAD dar. Der größte Unterschied zu MLCad besteht darin, dass innerhalb von LeoCAD vollständig in einer 3D-Ansicht modelliert werden kann.

Desweiteren besitzt das Programm seine eigene Basisdatenbank und arbeitet nicht nur mit der LDraw-Bibliothek. Somit ist LeoCAD eine völlig eigenständige Anwendung, die eine LDraw-Installation grundsätzlich nicht erfordert.

1) *Platzieren, bewegen und rotieren:* Snapping wird in LeoCAD zwar beim Einfügen von neuen Bausteinen durchgeführt, bei weiterem Bewegen oder Rotieren kann es aber ebenso wie in MLCad zu Deckungsfehlern kommen.

2) *Leistungsmerkmale:* Das auffälligste und gewöhnungsbedürftigste an LeoCAD ist wohl sicher, dass zum Auswählen, Bewegen und Rotieren von Teilen eigene Modi vorhanden sind. Wird ein Teil selektiert, muss, um es zu bewegen (rotieren), erst in den entsprechenden Modus gewechselt werden. Für eine neuerliche Auswahl ist wieder in den Selektionsmodus zu wechseln.

Große Modelle verursachen in LeoCAD hingegen keinerlei Probleme. Auch die Ansichtsteuerung ist sehr gut gelöst, so dass es während des Modellierens kaum nötig ist, die optional anzeigbaren Zusatz-Ansichten (insgesamt sind bis zu

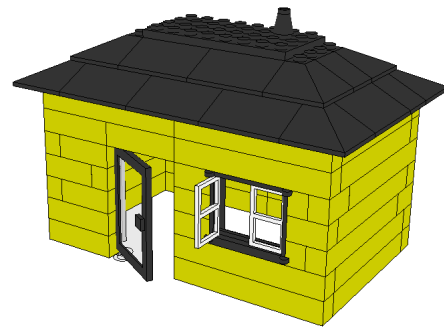


Abb. 7. Das selbe Hausmodell in der 3D-Modellieransicht von LeoCAD.

vier Ansichten gleichzeitig möglich) zu nutzen. Zusätzlich bietet LeoCAD einen Animationsmodus, in dem etwa animierte Bauanleitungen erstellt werden können.

Auch LeoCAD kann sowohl in Windows- wie auch Linux-Umgebungen eingesetzt werden.

C. SR 3D Builder

Der SR 3D Builder (SRB) ist ein noch relativ junges Projekt, das aber bereits über sehr viele nützliche Funktionen verfügt. Der SRB ist speziell für das LDR-Format entwickelt worden und nutzt auch die LDraw.org-Teile-Bibliothek. Wie der Name schon sagt, handelt es sich um ein Modellierungsprogramm im 3D-Modus. SRB benötigt eine Windows-Plattform, das .NET Framework sowie DirectX 9 [14].

Die Eingabesteuerung ist etwas gewöhnungsbedürftig und es ist leider auch noch keine ausreichende Dokumentation vorhanden, so dass manche Funktionen erst nach längerem Arbeiten entdeckt werden.

1) *Platzieren, bewegen und rotieren:* Wie auch in LeoCAD können hier Teile ohne Restriktionen platziert werden, obwohl eine Andock-Prüfung vorhanden ist. Translationen werden in SRB an einem 20 LRU breiten Raster ausgerichtet. Eine Detailausrichtungsfunktion ist zwar vorhanden, aber noch nicht ausreichend implementiert. Auch beim Rotieren gibt es Einschränkungen. So kann zwar um jede Achse rotiert werden, allerdings nur in Schritten von 90 Grad.

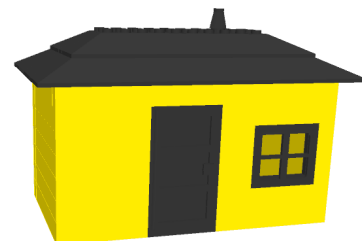


Abb. 8. Das LDR-Hausmodell in SR 3D Builder. Einige Objekte der LDraw-Bibliothek (Türen, Fenster) können nicht verwendet werden und müssen durch andere ersetzt werden.

2) *Leistungsmerkmale:* Auch bei größeren Modellen bietet der SRB schnelle Lade- und Bearbeitungszeiten und hat damit gegenüber anderen Werkzeugen einen entscheidenden Vorteil. Leider sind einige Programmfunktionen noch nicht vollständig implementiert: Neben der oben erwähnten Detailausrichtung wären das etwa das Gruppieren von Teilen, Bildausrichtungsprobleme beim Rotieren der Kamera oder Probleme beim Laden einiger LDR-Dateien (Abbildung 8).

D. LDDesignPad

LDDP ist ein Texteditor mit spezieller LDraw-Unterstützung. Er bietet neben Syntax-Highlighting auch eine ganze Reihe von nützlichen Shortcuts, wie etwa dem Einfügen von Standard-Header-Informationen, einer LDraw-Farbtabelle oder das Anzeigen des aktuellen Modells in einer externen Applikation, wie etwa MLCad oder LDView (IV-E.1).

E. Betrachtungsprogramme

Neben den Modellierwerkzeugen existieren auch reine Betrachtungsprogramme für LDR-Modelle.

1) *LDView:* Das am weitesten verbreitete Modell-Betrachtungstool ist LDView. Es basiert auf OpenGL und erlaubt es, LDraw-Modelle aus jedem Winkel zu betrachten. Weitere nützliche Anwendungen des Programmes sind:

- Steuerung der Lichteffekte
- Drahtmodellausgabe
- Ausgabe von Stereobildern
- Erstellen von Teile-Listen im HTML-Format
- Erstellen von Bildschirmschonern aus LDraw-Modellen

2) *Weitere Programme:* Auch LdLite und dessen verbesserte Nachfolgeapplikation LdGLite sind OpenGL-basierte Viewer für LDraw-Modelle. L3Lab ist ein weiterer LDR-Modellbetrachter und wird von LDraw.org offiziell empfohlen. L3Lab entstand aus dem Projekt L3P (Kapitel IV-F.1).

F. Rendering mit POV-Ray

Um die erstellten Modelle fotorealistisch darstellen zu können, bedarf es der Verwendung zusätzlicher Programme. POV-Ray ist ein Renderer, der Raytracing verwendet und Daten im POV-Format verarbeitet. Die Scene Description Language (SDL) dient dabei zum Beschreiben der zu erzeugenden Szenen [15]. Um LDR-Modelle mittels POV-Ray zu rendern, werden Zusatzprogramme benötigt, die eine Format-Konvertierung vornehmen.

1) *L3P:* Die Transformation der LDR-Daten in das SDL-Format erfolgt in mehreren Schritten, wodurch zwar längere Renderzeiten, dafür aber um ein Vielfaches kleinere POV-Dateien entstehen. Mit L3Paddon steht eine graphische Benutzeroberfläche zur Verfügung, da L3P ursprünglich kommandozeilen-orientiert implementiert war.

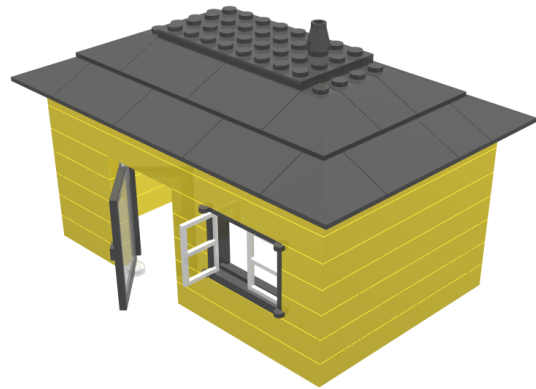


Abb. 9. Das zuvor mit MLCad erstellte Haus gerendert mit POV-Ray 3.6.

2) *L3Pub:* Im Zuge seiner Arbeiten an einem Buch [13] über LDraw entwickelte Kevin Clague L3Pub, eine Art Sammelprogramm, das die Erstellung von Raytracing-Bildern aus LDraw-Daten automatisiert. In L3Pub eingelesene LDR-Daten werden mit L3P in das entsprechende Format umgewandelt; anschließend wird POV-Ray aufgerufen. Zusätzlich können LDGLite oder LDView zum Rendern der Modelle ausgewählt werden. Sämtliche Einstellungen für die verwendeten Programme können direkt in der L3Pub-Oberfläche getätigt werden.

Weitere Features von L3Pub sind das Erstellen von gerenderten Einzelschritt-Bauanleitungen und Bauteilübersichten dieser Einzelschritte.

V. DER LEGO DIGITAL DESIGNER

Der Lego Digital Designer (LDD) ist das offizielle Tool zum Erstellen von Bauanleitungen und Modellen. Er ist als Freeware auf der Lego-Webseite in der aktuellen (Stand: Januar 2008) Version 2.3 zum Download verfügbar. Mithilfe des LDD können nicht nur Modelle erstellt werden, sondern diese auch über den Online-Shop als Einzelfertigungsstück bestellt werden.

Mit 763 verfügbaren Originalteilen ist der LDD zwar in seinen Gestaltungsmöglichkeiten gegenüber LDraw stark begrenzt, bietet jedoch sowohl eine leicht zu bedienende graphische Oberfläche als auch die Möglichkeit, offizielle Bauanleitungen und Modelle von der Webseite zu beziehen und zu bearbeiten.

Seinen Ursprung findet das LDD-Projekt in einem Wettbewerb, in dem dazu aufgerufen wurde, für das Computerspiel Lego-Creator eigene Modelle zu kreieren. Dazu wurde der Digital Designer bereitgestellt. Aufgrund des positiven Feedbacks und marketingtechnischer Überlegungen wurde das Projekt fortgeführt.

A. Unterstützte Dateiformate

Der LDD unterstützt drei verschiedene Dateiformate: Neben dem Verarbeiten des eigenen LXF-Formats kann auch mit LXFML- und LDR-Dateien gearbeitet werden. LXFML-Dateien folgen dabei dem gängigen XML-Standard. Die LDR-Unterstützung ist in der Version 2.3 allerdings noch sehr

unausgereift und nahezu unbrauchbar, da die meisten Modelle selbst dann nicht vollständig eingelesen werden können, wenn nur originale Lego-Teile verwendet wurden. Die Spezifikation des binären LXF-Formates wurde bisher nicht veröffentlicht.

B. Modellieren mit LDD

Das Modellieren geschieht nur mit Hilfe von Standard-Teilen und ist sehr dem realen Bauen mit Lego-Blöcken nachempfunden. Wie auch in den meisten anderen Editoren ist es im LDD nicht möglich, Teile in ihrer Form zu verändern. Auch können alle Teile nur in tatsächlich erhältlichen Farben verwendet werden. Da es sich um ein offizielles Produkt handelt, soll dieses auch repräsentativ das einzigartige Lego-Baugefühl vermitteln.

Es stehen dem Benutzer drei Standardmodi zur Verfügung: Lego-Factory, Lego-Mindstorms und Lego-Creator. Factory und Creator beinhalten nur Standard-Teile, im Mindstorms-Modus können auch die programmierbaren Zusatzteile verbaut werden. In jedem Modus ist sowohl ein freies Bauen als auch das Laden von vorgefertigten oder eigenen Modellen möglich. Für jedes erzeugte Modell wird automatisch eine animierte Bauanleitung generiert, die alle Schritte aufzeigt, die notwendig waren, um zur resultierten Szene zu gelangen.

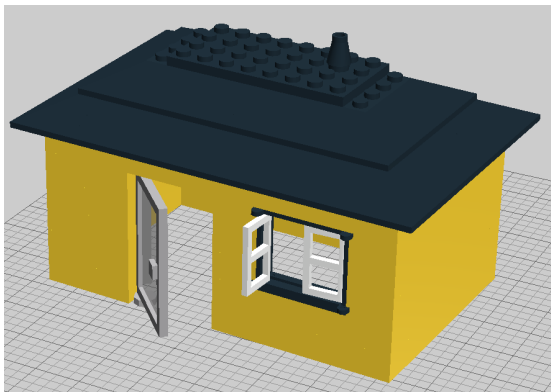


Abb. 10. Modell eines simplen Hauses, erzeugt mit LDD. Tür und Fensterflügel sind an die Rahmen angedockt und nicht mehr frei beweglich. Es wurden die gleichen Teile wie im MLCad-Beispiel verwendet.

C. Platzieren, bewegen und rotieren

Die überaus hilfreiche Snap-Funktion vermittelt ein echtes Lego-Baugefühl. Steine docken automatisch an anderen an, wenn man sie in deren Nähe platziert. Da der LDD aber nicht als umfassendes CAD- oder Grafik-Werkzeug konzipiert ist, sind grundlegende Funktionalitäten wie Rotation und Gruppierung von Objekten zwar vorhanden, aber wenig zufriedenstellend implementiert.

D. Leistungsmerkmale

Die Tatsache, dass keine eigenen erstellten Bausteine verwendet werden können, erweist sich für das Erstellen komplexerer Modelle als enormer Nachteil. Darüberhinaus ist der LDD sehr speicherlastig, die Performance ist selbst auf Systemen, die die Mindestanforderungen weit übertreffen, schon

bei kleinen Modellen eher schwach. Unterstützte Plattformen sind Windows und Mac. Als Zielgruppe der Software sind eher potentielle und tatsächliche Lego-Kunden zu sehen.

VI. WEITERE WERKZEUGE

A. BlockCAD

BlockCAD ist ein Freeware-Programm zum Modellieren mit Bausteinen. Es besitzt eine eigene Bibliothek, in der Original-Lego-Steinen nachempfundene und eigens kreierte Steine vorhanden sind (siehe Abbildung 11). Ein Hinzufügen weiterer Bausteine ist zwar möglich, allerdings nur in einem eigenen Format. Daher fehlt eine direkte Anbindung an LDraw. Das Programm ist weiteren Restriktionen unterworfen [16]:

- Das Verschieben bereits platzierter Objekte ist nur ungenügend realisiert.
- Konkave Teile können erzeugt werden, allerdings dürfen in diesen konkaven Bereichen keine anderen Objekte platziert werden.
- Objekte können nicht beliebig rotiert werden. Eine Ausrichtung der Teile, sodass die Noppen nicht nach oben zeigen, ist nicht möglich.

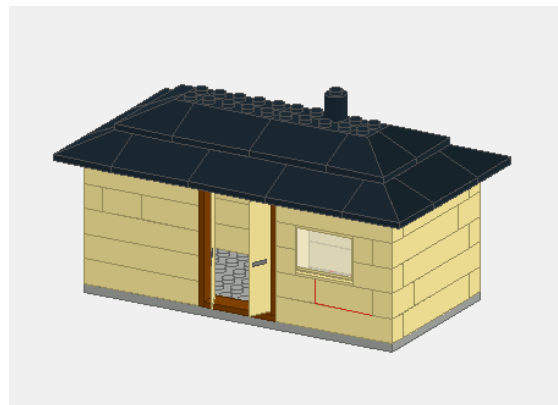


Abb. 11. Mit BlockCAD erstelltes Hausmodell.

B. Virtual Lego

Im Zuge der Suche nach Interaktionsmöglichkeiten in virtuellen Umgebungen entwickelten Stuerzlinger und Oh [6] das Programm Virtual Lego, ein simples Modellierwerkzeug, das auf der Lego-Grundidee basiert. Virtual Lego besteht aus zwei Teilprogrammen, die sich sehr ähnlich sind und nur darin unterscheiden, wie angrenzende Teile beim Verschieben von einzelnen Teilen behandelt werden.

Virtual Lego ist ursprünglich nicht dazu gedacht, größere Modelle aus Lego-Steinen zu erzeugen. Daher steht auch nur eine sehr geringe Auswahl an unterschiedlichen Bauteilen zur Verfügung (Abbildung 12). Das besondere an diesem Werkzeug ist allerdings, dass sich Teile in ihrer Form verändern lassen. Virtual Lego bietet daneben auch Snapping, eine ausgereifte Selektionstechnik und einen vollständigen 3D-Modelliermodus.

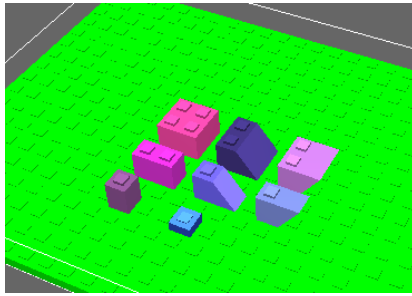


Abb. 12. Virtual Lego bietet nur eine sehr begrenzte Auswahl an Bauelementen, die allerdings in ihrer Form verändert werden können.

VII. EINSETZBARKEIT DER WERKZEUGE

Welche dieser Werkzeuge können nun für wissenschaftliche Zwecke von Nutzen sein? Dazu muss man vorerst Fragen darüber beantworten, wie das Ziel lautet, das mit dem gewählten Tool erreicht werden soll. Diese könnten etwa wie folgt lauten:

- Ist das Programm nutzbar, um technisch brauchbare Modelle zu erzeugen?
- Können Benutzer ohne Erfahrung mit CAD-Werkzeugen das Programm benutzen?
- Hält sich das Werkzeug an grundlegende Lego-Bau-Prinzipien wie Snapping?
- Sind Zusatzprogramme vorhanden?

Einen Überblick über die bisherigen Erkenntnisse soll die Tabelle I liefern. Virtual Lego (Kapitel VI-B) kann aus der Evaluierung ausgeschlossen werden, da der Einsatzbereich durch die Programm-Konzeption bereits festgelegt ist.

Fragestellung	MLCad	LeoCAD	SRB	LDD	BCAD
Technisch nutzbar	+	+	+	o	-
Unerfahrene Benutzer	-	-	-	+	+
Lego-Prinzipien	o	o	o	+	+
Zusatzprogramme	+	+	+	-	-

Tabelle I

ZUSAMMENFASSUNG DER BEWERTUNGSKRITERIEN FÜR DEN WISSENSCHAFTLICHEN EINSATZ.

A. CAD

Für den tatsächlichen Einsatz als CAD-Anwendung kommen von den ausgewählten Programmen nur jene in Frage, die auf LDraw basieren. Weder der LDD noch isolierte Werkzeuge wie BlockCAD können eine ähnliche Tiefe an Modellierfunktionalität bieten. Auch die bereits vorhandenen Zusatzprogramme sprechen eindeutig für eine LDraw-basierte Modellierung.

MLCad besitzt dabei gegenüber LeoCAD und SRB den entscheidenden Vorteil, sowohl eine leicht zu erlernende Eingabesteuerung als auch hilfreiche Zusatzfunktionalitäten vorzuweisen.

B. Usability-Studien

Vergleichende Usability-Studien für alternative Eingabesteuerungen in CAD- oder VR-Systemen bauen darauf auf, dass Benutzer, die in solchen Systemen keine Erfahrung vorzuweisen haben, das Bauen mit Lego-Steinen - sofern nicht bekannt - sehr schnell erlernen können. Daher ist ein anschließender Vergleich mit dem interaktivem Bauen sehr aussagekräftig. Hier hat vor allem eine Nachahmung des Verhaltens von Lego-Steinen eine besondere Bedeutung. Virtual Lego erfüllt diese Aufgabe nur in den Bereichen, für die es tatsächlich ausgelegt ist (Gruppierung, Bewegung von Gruppen). Auch der Digital Designer erweist sich hier als Alternative, erfüllt aber ebenso nicht alle gestellten Anforderungen. Von den in dieser Arbeit betrachteten Werkzeugen ist somit keines für einen wirklich umfassenden Einsatz geeignet.

C. Lernforschung

Der Einsatz von Lego-CAD-Software in der Lernforschung setzt eine relativ schnelle und einfache Erlernbarkeit voraus. Auch die Verbindung hin zur realen Konstruktion und ein daran angelehntes Bauverhalten spielen eine große Rolle.

Von den vorgestellten Werkzeugen erweist sich letztlich keines als in diesem Bereich optimal einsetzbar, da keines die hohen Anforderungen erfüllen kann. Am ehesten geeignet erscheint aus derzeitiger Sicht die Technologie Lego-Mindstorms [17], die allerdings nicht dem CAD-Bereich zuzurechnen ist, da sämtliche Konstruktionsarbeiten mit realen Steinen erfolgen.

Einsatzgebiet	MLCad	LeoCAD	SRB	LDD	BCAD	VL
CAD	+	+	+	o	-	-
Usability	-	-	-	o	-	+
Lernforschung	-	-	-	o	-	o

Tabelle II

LEISTUNGSSTÄRKE DER BEHANDELTEN PROGRAMME IN AUSGEWÄHLTEN BEREICHEN.

VIII. ZUSAMMENFASSUNG UND AUSBLICK

Lego hat den Schritt in die Computergraphik bereits getätigt, auch wenn viele mögliche Einsatzfelder noch nicht ausreichend genutzt werden (können). Vor allem die Verbindung der Erfahrungen im Bauen mit realen und virtuellen Bausteinen kann wichtige Informationen darüber liefern, wie man Eingabesteuerung bestmöglich (im Sinne von: Benutzerorientiert) realisieren kann.

LDraw bietet durch seine große Vielfalt und die reichlich vorhandenen Werkzeuge äußerst umfangreiche Einsatzmöglichkeiten, scheint aber für Benutzer ohne CAD-Vorwissen ungeeignet, da die meisten Programme ein gewisses Grundverständnis voraussetzen. Daher fehlt es in vielen wissenschaftlichen Bereichen noch an den entsprechenden Werkzeugen, um das einzigartige Produkt Lego richtig einzubinden.

Lego Mindstorms ist eine hervorragende Lösung für dieses Problem im Bereich der Lernforschung, aber letztlich zu technisiert, um in allen Altersgruppen einsetzbar zu sein. Hier könnte der Ansatzpunkt für eine getrennte Lösung aus

realer Konstruktion und Lego-basierten Programmen weitere Fortschritte bringen.

REFERENZEN

- [1] M. Resnick, "Computer as paint brush: Technology, play, and the creative society," in *Play = Learning: How play motivates and enhances children's cognitive and social-emotional growth*, K. Hirsh-Pasek, R. M. Golinkoff, and D. G. Singer, Eds., 2006. [Online]. Available: <http://llk.media.mit.edu/papers.php>
- [2] M. Resnick, F. Martin, R. Sargent, and B. Silverman, "Programmable bricks: toys to think with," *IBM Syst. J.*, vol. 35, no. 3-4, pp. 443-452, 1996. [Online]. Available: <http://www.research.ibm.com/journal/sj/353/sectionc/martin.html>
- [3] M. Resnick, "Behavior construction kits," *Commun. ACM*, vol. 36, no. 7, pp. 64-71, 1993. [Online]. Available: <http://web.media.mit.edu/~mres/papers/BCK/BCK.pdf>
- [4] "Lego mindstorms," 2008. [Online]. Available: <http://mindstorms.lego.com/Overview/>
- [5] H. Baradaran and W. Stuerzlinger, "A comparison of real and virtual 3d construction tools with novice users," in *CGVR'06*, York University, Toronto, ON, June 2006. [Online]. Available: <http://www.cse.yorku.ca/~wolfgang/publications.html>
- [6] J.-Y. Oh and W. Stuerzlinger, "A system for desktop conceptual 3d design," *Virtual Real.*, vol. 7, no. 3-4, pp. 198-211, 2004. [Online]. Available: <http://www.cse.yorku.ca/~wolfgang/publications.html>
- [7] T. Courtney, "Ldraw history," online, May 2003. [Online]. Available: <http://www.ldraw.org/Article11.html>
- [8] S. Bliss, O. Pobursky, W. Tschager, and W. Howard, "Ldraw.org file format version 1.0.0," September 2003. [Online]. Available: <http://www.ldraw.org/Article218.html>
- [9] O. Pobursky, "Ldraw.org color chart," September 2003. [Online]. Available: <http://www.ldraw.org/Article93.html>
- [10] S. Bliss and O. Pobursky, "Ldraw.org mpd language extension," online, May 2003. [Online]. Available: <http://www.ldraw.org/Article47.html>
- [11] S. Bliss, O. Pobursky, W. Tschager, and W. Howard, "Ldraw.org category and keywords language extension," November 2004. [Online]. Available: <http://www.ldraw.org/Article340.html>
- [12] M. Lachmann, "Mlcad," April 2006. [Online]. Available: <http://www.lm-software.com/mlcad/>
- [13] K. Clague, M. Agullo, and L. C. Hassing, *LEGO Software Power Tools, With LDraw, MLCad, and LPub*. Syngress Publishing, 2003.
- [14] "Sr 3d builder. a new way to build your lego models." [Online]. Available: <http://staff.polito.it/sergio.reano/>
- [15] "Pov-ray: Documentation." [Online]. Available: <http://www.povray.org/>
- [16] A. Isakson, "Blockcad technical info," 1998. [Online]. Available: <http://web.telia.com/~u16122508/technical.htm>
- [17] H. Patterson-McNeill and C. L. Binkerd, "Resources for using lego mindstorms," in *Proceedings of the seventh annual consortium for computing in small colleges*. USA: Consortium for Computing Sciences in Colleges, 2001, pp. 48-55.