

Die UB in 3D oder: wie kommt der Benutzer zum Buch?

Jochen Barth

Das Gebäude der Universitätsbibliothek Heidelberg, „eines der dienstältesten universitären Bibliotheksgebäude“¹ wurde ursprünglich für 800 studentische Benutzer ausgelegt. Die Räumlichkeiten waren leicht überschaubar: dem Benutzer war nur der südliche Teil des Gebäudes zugänglich. Der übrige Teil des Gebäudes war Magazin. Mit steigender Zahl der Benutzer (2010: ca. 41.500 aktive Benutzer, 6.185 Besucher/Tag) wurde es erforderlich, weitere Flächen zugänglich zu machen.

```
#include „colors.inc“

camera { // Angaben zum Betrachter
  location <300, 160, 0> // 300 cm östlich, 160 „Augenhöhe“
  look_at < 0, 100, 50 > // Blick nach Westen (ca.)
}

light_source { // Angaben zur Lichtquelle
  < 0, 1000000, -1000000 > // Sonne steht im Süden
  color White
}

plane { // Boden
  < 0, 1, 0 >, 0 // Normalenvektor zur Ebene, Versatz
  pigment { color Gray }
}

sphere { // Kugel
  < 0, 130, 0 >, 50 // Position, Radius 50 cm

  pigment { color Cyan }
}
```

Text 1: Beispiel Szenenbeschreibung in Pov-Ray

Durch die im Jahr 1977 erfolgte Umnutzung der Magazingeschosse zu Freihandbereich und Lesesaal, der - seit 1996 - zunächst beschränkten Erweiterung der UB in das sog. Triplexgebäude hinein und der seit 2009 erfolgten „Triplex-Norderweiterung“² wurden

die Räumlichkeiten für die Benutzer deutlich unübersichtlicher.

Im Opac wurden zur Orientierung zwar bereits zweidimensionale Pläne angezeigt, diese waren jedoch recht einfach gestaltet und eigneten sich wegen der gegebenen Komplexität nicht mehr zur Orientierung. So entstand die Idee, die UB schematisch in einem dreidimensionalen Modell zu erfassen und Bilder mit hervorgehobenen Regalen von diesem Modell zu erzeugen, um dem Benutzer deren Lage sichtbar zu machen.

Da sich der Preis für kommerzielle Produkte nach den lokalen Gegebenheiten und dem Anpassungsaufwand richtet, waren solche Produkte im Falle der UB Heidelberg finanziell unattraktiv.

Dem Autor war aus früheren Linux-Tagen noch das Programm „Pov-Ray“³, ein sog. Raytracer bekannt - ein Programm, das aus einer Szenenbeschreibung („Modell“) ein Bild errechnet. Die Szene wird in einer Textdatei mit der Endung .pov beschrieben (siehe Beispiel in Text 1).

Grafische Benutzeroberflächen für Pov-Ray sind wegen der drei Dimensionen relativ schwierig zu bedienen. Da die Szenenbeschreibungssprache vergleichsweise einfach ist, wird hier für die Eingabe von Böden, Wänden und Mobiliar Vektor-Zeichenprogramm „Inkscape“⁴ verwendet. Dieses eignet sich aufgrund seiner einfachen Bedienung, seinem offenen Datenformat SVG (XML) sowie der strukturelle Ähnlichkeit von SVG zur Szenenbeschreibungssprache von Pov-Ray bestens für diese Aufgabe.

Die Software Inkscape ist Open Source, Pov-Ray ist kostenfrei⁵.

Um die Vorteile beider Programme zu verbinden, hat der Autor das Programm „svg2pov.pl“ entwickelt, das .svg-Dateien in .pov-Dateien umwandelt - gesteuert durch eine Ansichten-Konfigurationsdatei (siehe auch Text 3). In dieser ist festgelegt, in welcher Ansicht bzw. Bild welche Objekte hervorgehoben werden sollen, z. B. Schließfachbereiche, Regale, Wegweiser etc.

³ Wegen der SMP-Unterstützung wird Version 3.7 beta RC3 empfohlen.

⁴ Siehe <http://www.inkscape.org>

⁵ Siehe <http://www.povray.org/povlegal.html>

¹ <http://www.ub.uni-heidelberg.de/allg/bau/>

² <http://www.ub.uni-heidelberg.de/allg/bau/>

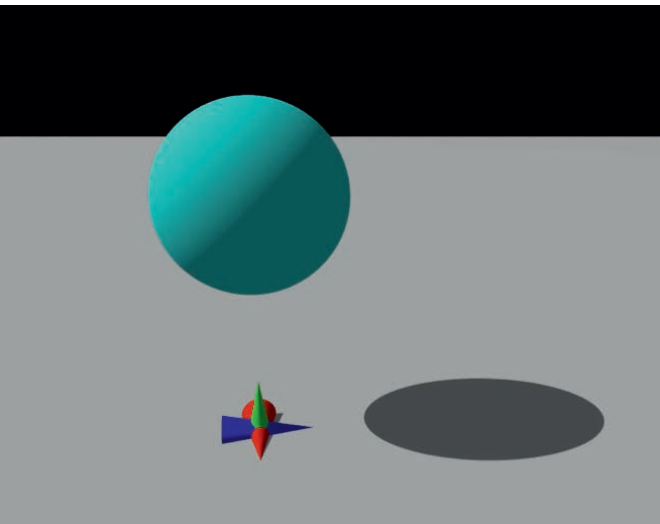


Abbildung 1: Von Pov-Ray aus der Beispiel-Szene berechnetes Bild

So entsteht mittels `svg2pov.pl` aus dem SVG-Modell und den Ansichten-Definitionen je Ansicht eine `.pov`-Datei, aus der Pov-Ray ein Bild errechnet.

Eingabe in Inkscape

siehe auch Abb. 3

Inkscape speichert die Objekte in der `.svg`-Datei in der Einheit „Pixel“ („px“) – daher empfiehlt sich diese Einstellung auch für die Anzeige in Inkscape. Für die Eingabe gilt dann z. B. 1 px entspricht 1 cm.

Inkscape verfügt nur über zwei der drei für die Beschreibung erforderlichen Dimensionen. Die Angabe der Höhe eines Objektes erfolgt daher im Beschreibungsfeld (siehe Abb. 2) der Objekteigenschaften dieses Objektes mit „`height1: Zahl; height2: Zahl;`“. Diese Höhenangaben beschreiben die Ausdehnung eines Objektes in Richtung Y-Achse (Höhe) relativ zur Inkscape-Ebene, in der sich das Objekt befindet (s. u.). Beispiel: „`height1:100; height2:200;`“ würde im Falle eines Inkscape-Quadrates einen dreidimensionalen Quader mit der Höhe von 100 cm beschreiben (200–100 cm), der 100 cm über dem Geschossboden schwebt.

Die Eingabe eines Gebäudes erfolgt geschossweise, mit je einer Ebene für den gescannten Grundriss, immobile (z. B. Wände) und mobile (Regale etc.) Objekte. Diese Aufteilung ist nicht zwingend, erleichtert aber die Eingabe der mobilen Objekte, da die Sperrung der immobilen Ebenen versehentliche Änderungen derselben verhindert.

Weil Ebenen in Inkscape kein Beschreibungsfeld haben, erfolgt die Angabe der Höhe eines Geschosses durch Benennung der Ebene mit dem Suffix „+“ oder „-“ gefolgt von der Höhe, also z. B. „OG1+280“.



Schritt 1: Aussparungs-Umriß anlegen („Aussparung für Wendeltreppe“)

Schritt 2: Bodenplatte und Aussparung markieren

Schritt 3: Menü „Pfad“ → „Differenz“

Text 2: Inkscape: Differenz von Objekten erstellen

Voraussetzung für die Eingabe der Geschosse ist ein eingescannter oder bereits digital vorliegender Grundriss. Zunächst wird der Grundriss auf die Boden-Ebene mit Linienzügen oder Bézier-Kurven „durchgepaust“. Aussparungen in der Bodenplatte für Treppenhäuser und Aufzüge können in Inkscape über den Befehl „Differenz“ im Menü „Pfad“ erzeugt werden. Dann erfolgt die Eingabe der Wände, Treppen und Aufzüge. Für die Wände empfiehlt es sich, eine Kopie der Bodenplatte anzulegen und die Räume mit dem „Differenz“-Befehl (siehe auch Text 2) herauszuschneiden. Für den Benutzer unzugängliche Räume wurden als „Wand“ belassen und nicht einzeln herausgearbeitet.

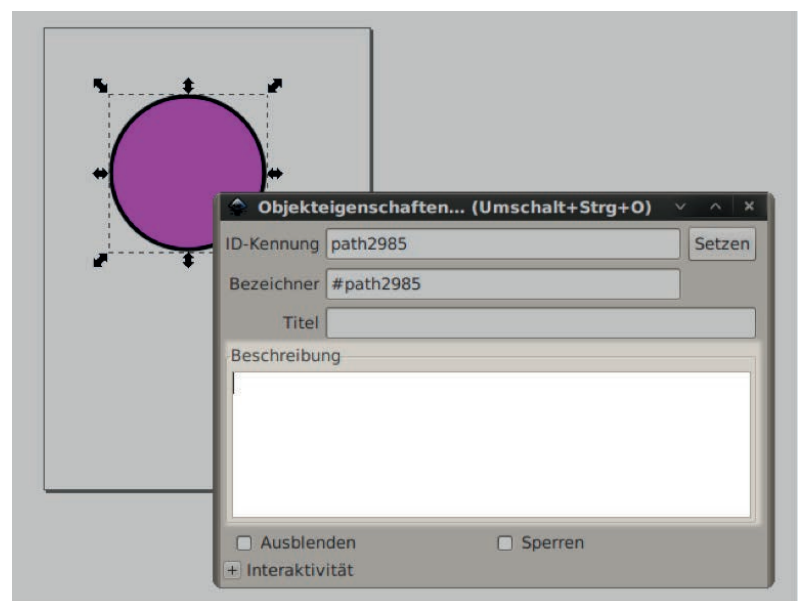


Abbildung 2: Inkscape: Beschreibungsfeld eines Objektes

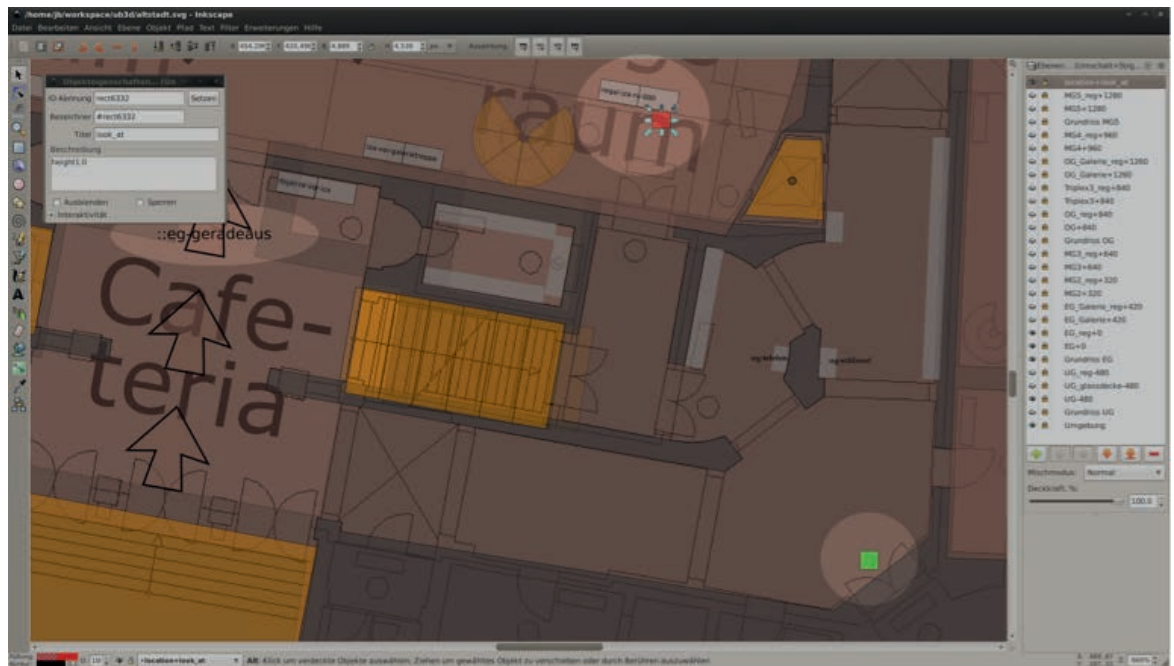


Abbildung 3: Eingabe in Inkscape; Quelle des Lageplans: ap88

Zuletzt erfolgt die Eingabe der immobilien Objekte. Für Treppen und Regalreihen gibt es vom Autor entwickelte Hilfsprogramme, die solche generieren. Für Treppen muss die Höhendifferenz und die Zahl der Stufen vorgegeben werden, bei geraden Treppen zusätzlich die Breite und Tiefe, bei gewendelten Treppen die Winkeldifferenz und der Außen- und Innenradius. Für Regalreihen ist die Angabe der Anzahl der Reihen, deren Tiefe, deren Gesamtbreite (oder der Abstand zwischen den Regalen) sowie der Grundbeschriftung zur automatischen Nummerierung erforderlich. Wegen der Größe der Objekte im endgültigen Bild empfiehlt es sich, die Objekte zu vereinfachen und zu vereinheitlichen. Die Zahl der Treppenstufen sind im UB-Modell des Autors nur ein Drittel der tatsächlichen. Präzision ist nicht erforderlich - aus der für die Betrachtung des Gebäudes erforderlichen Entfernung sind kleine Differenzen nicht zu erkennen. Die Benennung von Objekten für die Hervorhebung erfolgt durch Gruppierung des Objekte (oder Objektgruppe) mit einem Text-Objekt, das den Namen des Objektes mit führendem Doppelpunkt beinhaltet. Text-Objekte ohne führenden Doppelpunkt erscheinen als solcher im Bild.

Ansichten-Konfiguration

In der für `svg2pov.pl` benötigten Ansichten-Konfigurationsdatei (Text 3) werden die auszugebenden Ansichten bzw. Bilder definiert und jeweils die hervorzuhebenden Objekte und Wegweiser. Die einzelnen Ansichten-Definitionen können außerdem zusätzliche Angaben wie den Signaturbereich,

den die hervorzuhebenden Regale abdecken oder eine Wegbeschreibungen in Textform enthalten. Diese Zusatzangaben werden von `svg2pov.pl` nicht ausgewertet. Die Zusatzangaben zum Signaturbereich beispielsweise müssen vom OPAC (oder einem zwischengeschalteten CGI-Script) ausgewertet werden, damit dieser aus der Menge der verfügbaren Ansichten die richtige ermitteln kann.

Für die Erstellung der `.pov`-Datei ermittelt `svg2pov.pl` automatisch, welche Ebenen über den hervorzuhebenden Objekten liegen und blendet diese aus. Für Wegweiser muss die Benennung mit zwei aufeinanderfolgenden Doppelpunkten beginnen - als Kennzeichnung dafür, dass diese bei der automatischen Ausblendung von Ebenen nicht berücksichtigt werden sollen.

`svg.pov.pl` erstellt dann aus der `.svg`- und Ansichten-Konfigurationsdatei je Ansicht eine `.pov`-Datei, die dann von Pov-Ray berechnet und als Bild gespeichert wird (Text 4).

Inkscape kennt leider keine Formatvorlagen für gleichartige Objekte, d. h. es fehlt die Möglichkeit, die Farbe für z. B. alle Regale mit wenigen Handgriffen zu ändern. Daher gibt es in in dem Modul „preferences.pm“ eine Ersetzungstabelle, die es ermöglicht, in Inkscape festgelegte Farben durch andere zu ersetzen. In dieser Ersetzungstabelle können auch Pov-Ray-spezifische Texturen und Materialien - z. B. Glas für das Erdgeschoss bei darunter liegendem Kellergeschoss - verwendet werden.

```
# Kommentare werden mit „#“ eingeleitet
# Ansichten-Definitionen werden mit „*“
# eingeleitet.
```

* Informationszentrum Präsenzbestand Allgemeines (IZA AI)

```
:regal-iza-ru-000 # hervorgehobenes Regal
:eg-geradeaus # hervorgehobener Wegweiser
```

```
# Zusätzlicher Angaben,
# werden nicht von svg2pov.pl
# ausgewertet:
# für OPAC: Signaturen zu dieser Ansicht
```

Signatur-von=0900 IZA AI*

Signatur-bis=0900 IZA AI*

```
# zusätzliche einzublendender Text unterhalb
des
```

```
# Bildes – für die Anzeige im OPAC.
```

Zugang=Über das Foyer gelangen Sie in das Informationszentrum im Erdgeschoss. Dort finden Sie diese Nachschlagewerke hinter der Informationstheke.

Text 3: Auszug aus der Ansichten-Konfigurationsdatei „magazingeschoss1-iza.cfg“

```
./svg2pov.pl altstadt.svg magazingeschoss1-iza.cfg
```

Erzeugt die Ausgabedatei

```
altstadt/Informationszentrum%20Präsenz-
bestand%20Allgemeines%20%28IZA%20AI%29 .pov
```

(Dateiname entspricht Ansichtenbezeichnung in URI-Codierung, weil Pov-Ray keine Leerzeichen in Dateinamen erlaubt)

Anschließend wird die Szene von Pov-Ray berechnet:

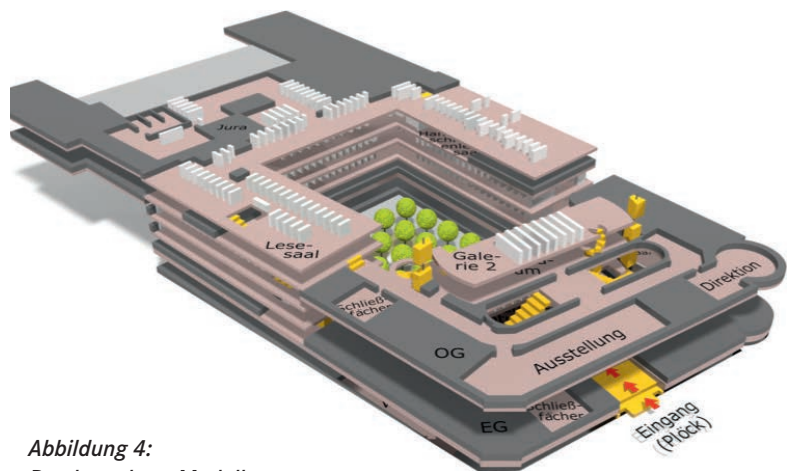
```
povray altstadt/Informationszentrum%20Präsenz-
bestand%20Allgemeines%20%28IZA%20AI%29.pov
```

Erzeugt das Bild altstadt/Informationszentrum%20Präsenzbestand%20Allgemeines%20%28IZA%20AI%29.png (Abb. 4)

Text 4: Erzeugung und Berechnung der .pov-Dateien

Markiert ist die Position des Blickpunktes – das rote Quadrat. Objekteigenschaft „Titel“: „look_at“

Grünes Quadrat: Position des Betrachters, Objekteigenschaft „Titel“: „location“. **I**



*Abbildung 4:
Das komplette Modell
der UB Heidelberg*



Jochen Barth

IT

Universitätsbibliothek

Heidelberg

barth@ub.uni-heidelberg.de

Sollen die .svg-Daten später für WebGL verwendet werden, müssen die Bezier-Kurven trianguliert, d. h. in viele kleine Dreiecke umgerechnet werden. Die Software ist unter Linux und Windows lauffähig und beim Autor erhältlich.