VRML – X3D

VRML ist die Abkürzung für "Virtual Reality Modeling Language". Das ist ein Fileformat zur Beschreibung interaktiver und multimedialer 3D-Welten im Internet. Die erste von Silicon Graphics entwickelte Version VRML 1.0 wurde in Zusammenarbeit mit Sony und anderen Entwicklern zur Version VRML 2.0 erweitert. Neben geringen Veränderungen in der Sprachstruktur wurden insbesondere die Möglichkeit von Animationen, Multimediaeffekten und der Aufruf von externen Scripts hinzugefügt. Diese Version wurde in nur geringfügig geänderter Form 1997 als internationaler Standard ISO/IEC 14772-1:1997 beschlossen und wird kurz als VRML97 bezeichnet. Mit ISO/IEC:14772-2:2002 wurde dieser Standard um das „External authoring interface (EAI)“ erweitert, das Schnittstellen von VRML zu anderen Programmiersprachen regelt. Derzeit durchläuft eine Weiterentwicklung X3D („Extensible 3D“) den Standardisierungsprozess. Die weitere Entwicklung wird vom Web3D Consortium koordiniert. Die jeweils aktuellen Informationen findet man auf <http://www.web3d.org> Da X3D insbesondere wegen des noch laufenden Standardisierungsprozesses noch nicht so verbreitet ist, behandeln wir hier nur VRML.

In VRML erfolgt die Beschreibung durch eine Hierarchie von sogenannten Knoten (Nodes), die in Feldern wieder Knoten enthalten, So ein Knoten beschreibt ein Geometrieobjekt als eine Gruppe von Objekten (die dann einer Transformation unterworfen werden können), Material- und Oberflächeneigenschaften sowie Sound. Seit VRML 2.0 gibt es auch Script-Knoten, die Scripts ausführen und externe Scripts aufrufen können. Um Animationen ausführen zu können, gibt es Knoten die Ereignisse melden können, etwa Sensoren oder Timer. Diese werden dann mit einem ROUTE Befehl zu Eingabeparametern (etwa Größe eines Translationsvektors) anderer Knoten weitergeleitet.

Ein ganz einfaches VRML-File, das einen Würfel generiert lautet:

```
#VRML V2.0 utf8

Shape {
  geometry Box {}

  appearance Appearance {
    material Material{
    }
  }
}
```

Die Werte für appearance sind zwar Leer, aber bei weglassen kommt es manchmal zu Darstellungsschwierigkeiten.

Die Möglichkeit den Standpunkt des Betrachters zu ändern und das Schattieren (Rendern) erfolgt dann automatisch durch einen VRML-Viewer, ein Programm zum Betrachten von VRML-Welten, das als Plug-In in den WWW-Browser integriert ist.

Software und weitere Informationen:

Plug-In zum Anzeigen von VRML-Files:

Cortona: <http://www.parallelgraphics.com/products/cortona>

Weitere Informationen:

WEB3D Consortium: <http://www.web3d.org>

Tutorial:

Lighthouse Interactive VRML Tutorial: <http://www.lighthouse3d.com/vrml/tutorial>

Ein interessanter Link:

Virtuelle Modelle: <http://www.uni-kl.de/AG-Leopold/VRML>

Ein etwas komplexeres VRML-File, das mittels eines Sensors die Mausbewegung abfragt, und danach dementsprechend die Kugel verschiebt, sieht so aus (Groß- und Kleinschreibung muss beachtet werden!):

```
#VRML V2.0 utf8

# In der ersten Zeile steht die Information über das Datenformat
# Alles was in einer Zeile hinter # steht ist ein Kommentar

NavigationInfo{
    type "EXAMINE"
}

# Knoten der Eigenschaften und Geometrie einer Kugel festlegt
Shape {
    geometry Box {}
    # Definition von Farbe, Material etc.
    appearance Appearance{
        material Material{
            diffuseColor 1 0 0
        }
    }
}

# Knoten der eine Gruppe von Elementen zusammenfasst,
# die einer Transformation unterworfen werden können.
# Die Vergabe eines Namens mit DEF kugel
# ist für die folgende Animation nötig

DEF kugel Transform {
    children[
        Shape{
            appearance Appearance{
                material Material{
                    diffuseColor 0 0 1
                }
            }
            geometry Sphere{
                radius 1
            }
        }
    ]

    # Hier wird ein Sensor definiert, der die Bewegung der
    # Maus in einer Ebene registriert

    DEF sensor PlaneSensor{
        minPosition 0 -2
        maxPosition 0 2
        offset 0 2 0
    }

    ]

    # Transformation aller children Objekte des Knotens
    translation 0 2 0
}

# Die Mausbewegung wird registriert und dementsprechend wird
# die Translation geändert

ROUTE sensor.translation_changed TO kugel.set_translation
```