

Praktikum zur Computergraphik

VRML-Miniprojekt

Ziel des Miniprojekts ist es, Erfahrungen mit einer etwas größeren Aufgabe in der Computergraphik zu sammeln und auch eine etwas komplexere Szene "mit allem drum und dran" aufzubauen. Schließlich ist es aufgrund der Dynamik auf dem Gebiet der Computergraphik wichtig, sich selbst zusätzliches Wissen anzueignen und sich selbstständig in ein Programm, eine Sprache oder eine API einzuarbeiten. Auch dies soll hier als wesentliches Ziel geübt werden: der Stoff der Vorlesung allein genügt nicht, um die Aufgaben zu bearbeiten - Eigeninitiative ist also nötig.

Das VRML-Miniprojekt darf nicht in Gruppen durchgeführt werden

Aufgabe V.1 (zu Teil A der Vorlesung)

Ziel der Aufgabe ist es, sich mit der VRML-Notation vertraut zu machen. Außerdem soll Erfahrung mit VRML-Werkzeugen gesammelt werden: VRML-Browser wie Cortona oder VRML-Editoren wie VRML-Pad. Zur Anzeige von VRML-Szenen benötigt man lediglich einen VRML-Browser. Diese gibt es von verschiedenen Herstellern und können in der Regel kostenlos aus dem Internet bezogen werden, z.B. Cortona bei www.parallelgraphics.com

Gegeben ist folgende VRML-Szene:

```
#VRML V2.0 utf8
Transform{
  rotation      1 1 1 0.785
  children [
    Transform{
      rotation    0 0 1 0.785
      translation 1 0 -10
      children [
        Shape{
          appearance Appearance{
            material Material {}
          }
          geometry Sphere {
            radius      2.4
          }
        }
      ]
    }
  ]
}
Shape{
  appearance Appearance{
    material Material {}
  }
  geometry Box {
    size 1 4 2
  }
}
```

- Lassen Sie sich die Szene anzeigen, indem Sie *meinErstesVRML.wrl* in einem VRML-Browser laden.
- Zeichnen Sie einen Szenengraphen, der nur Transform-Nodes und Shape-Nodes enthält

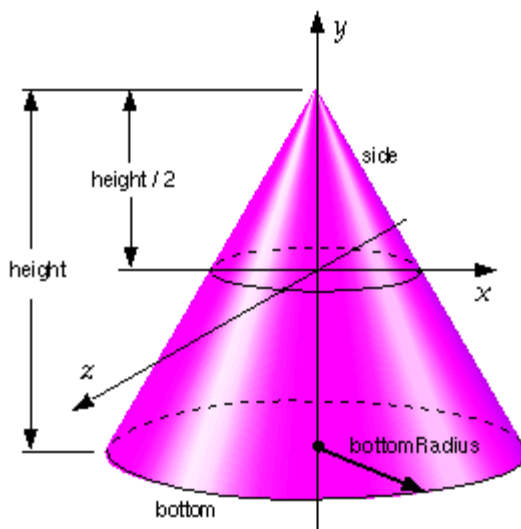


- (c) Verändern Sie die Parameter von **rotation** bzw. **translation** in beiden Transform – Nodes. Auf welche Objekte wirken sich die Veränderungen jeweils aus? Finden Sie heraus, was das Feld **scaleOrientation** im Transform Node bewirkt.
- (d) (Zusatzaufgabe) Versuchen unter Verwendung einer VRML-Dokumentation herauszufinden, wie man die Kugel grün und den Quader orange einfärben kann.
- (e) (Zusatzaufgabe) Zeichnen Sie den vollständigen Szenengraphen

Aufgabe V.2 (zu Teil B.1 der Vorlesung)

Ziel der Aufgabe ist es, die Verwendung von IndexedFaceSets und die dabei auftretenden Probleme mit dem richtigen Umlaufsinn an einem Beispiel noch einmal zu illustrieren. Ziel der Aufgabe ist es auch, die Verwendung des Szenengraph-Konzepts einzuüben, das ein sehr wichtiges Konzept in der Computer Graphik ist und nicht nur in VRML Anwendungen. Schließlich geht es um den DEF USE und PROTO Mechanismus von VRML.

- (a) Laden Sie die Datei *meinZweitesVRML.wrl*. Sie stellt ein (unvollständiges) Dreiecksprisma dar. Skizzieren Sie das Prisma.
- (b) Beim Betrachten aus der Standardblickrichtung des Browsers (welche ist das?) sehen Sie gar nichts (warum?). Wenn Sie das Objekt im Study-Modus drehen, sollten Sie zwei Vierecke und den dreieckigen Boden des Prismas sehen. Entfernen Sie # vor **solid FALSE** und sehen Sie in der VRML-Referenz nach, warum Sie jetzt etwas anderes sehen.
- (c) Ergänzen Sie die übrigen beiden Seitenflächen des Prismas (1 Dreieck und 1 Viereck) im Feld **coordIndex**.
- (d) Positionieren Sie einen Kegel (VRML-Node: **Cone**) so in der Szene, dass Kegel und Prisma sich nicht schneiden – nur die Spitze des Kegels soll genau den Mittelpunkt des Vierecks BCFE berühren – dabei soll der Grundkreis des Kegels parallel zum Viereck liegen. Hinweis: in seinem lokalen Koordinatensystem ist der Cone von VRML folgendermaßen definiert:.



- (e) (Zusatzaufgabe) Fügen Sie ein weiteres Prisma in Ihre Szene ein. Es soll die Höhe 2 (statt 1) haben und so verschoben werden, dass die Ecke E im Punkt (10,10,1) liegt. Dazu könnte man den Code des Prismas in einen passenden Transform-Knoten kopieren, aber es geht ohne Kopie mit dem Schlüsselwort **USE**.



- (f) (Zusatzaufgabe) Das eine Prisma soll nun grün, das andere rot gefärbt werden, ohne die Beschreibung des Prismas zu duplizieren. Hinweis: Erstellen mittels **PROTO** einen eigenen Node namens **Prisma** und geben Sie diesem ein Field namens **farbe**.

Aufgabe V.3 (zu Teil B.1 und B.2 aus der Vorlesung)

Nach den Vorübungen soll nun das Miniprojekt selbst gestartet werden: Skifahrer an einem Berghang. Dazu soll zunächst etwas modelliert werden: ein Berghang, Tannen, ein Schleplift mit Kassenhäuschen und ein Skifahrer.

- (a) Legen Sie eine Datei mit Namen an, wie es in den Bewertungsrichtlinien beschrieben wurde.
- (b) Erstellen Sie das Objektmodell eines Skifahrers mit Skiern an den Füßen. Verwenden Sie IndexedFaceSets für die Geometriebeschreibung des Skifahrers. Setzen Sie das Fahrzeug unter Verwendung von Transform Nodes zusammen. Benennen Sie die Knoten dabei und schauen sich dazu den DEF USE Mechanismus von VRML an. Für beide Ski soll ein einziger Quader (in VRML: Box) verwendet werden. Der Skifahrer soll insgesamt aus nicht mehr als 50 Flächen bestehen – für den Kopf kann man eine Kugel (in VRML: Sphere) verwenden. Der Skifahrer trägt einen roten Anzug.
- (c) Erstellen Sie unter Verwendung von Grundprimitiven den Berghang. Er besteht aus einer weißen geneigten Grundfläche (kann als Box modelliert werden) und einer weißen geraden Fläche (dem Auslauf), zehn Bäumen (kann aus Kegel und Zylinder modelliert werden), einem Schleplift mit fünf Säulen und ein Kassenhäuschen am Schleplift, das auf der geraden Fläche steht. Platzieren Sie den Skifahrer oben auf dem Hang, so dass die Skier parallel zum Hang liegen.
- (d) Erleichtern Sie das Erstellen mehrerer Skifahrer: definieren Sie einen Szenengraphen-Knoten-Typ mit Name Skifahrer in einer externen Datei (Stichwort: „Proto“ und „ExternProto“). Drei weitere Skifahrer sollen noch oben am Hang stehen. Jeder Skifahrer hat eine andere Farbe.

Aufgabe V.4 (zu Teil C aus der Vorlesung)

Der VRML-Browser bietet schon die Möglichkeit, z.B. mit der Maus, die Position und Ausrichtung der virtuellen Kamera zu verändern. Man kann aber mit Viewpoints schon feste Kamerapositionen festlegen.

- (a) Legen Sie einen Viewpoint fest, der den Berghang von oben aus der Vogelperspektive zeigt
- (b) Legen Sie 4 weitere sinnvolle Viewpoints Ihrer Wahl fest
- (c) Erweitern Sie den Knoten-Typ Skifahrer so, dass am Kopf jedes Skifahrers eine Kamera angebracht ist. Diese soll sich mitbewegen können und die First-Person-Perspective zeigen. Jeder Skifahrer muss dabei einen eigenen Namen haben.

Aufgabe V.5 (zu Teil D aus der Vorlesung)

Das Verwenden von Beleuchtung ist ein wesentliches Element, um Szenen in der Computergraphik gut aussehen zu lassen.



- (a) Hängen Sie Scheinwerfer oben an die Säulen des Schlepplifts, die nach unten den Berghang mit grünem Licht anstrahlen.
- (b) (Zusatzaufgabe) Auf dem Berghang soll man die Lichtkegel sehen.

Aufgabe V.6 (zu Teil E aus der Vorlesung)

- (a) Ihr Berghang schwebt derzeit noch in einem schwarzen Raum. Fügen Sie einen schöneren Hintergrund ein.
- (b) Die Scheinwerfer soll man über einen einzigen Schalter in der Szene ein- und ausschalten können. Der Schalter selbst kann als einfache Kugeln realisiert werden. Hinweis: verwenden Sie ein Skript, um sich den Zustand der Schalter zu merken.

Aufgabe V.7 (zu Teil E aus der Vorlesung)

- (a) Animieren Sie die Abfahrt des roten Skifahrers: der Skifahrer soll in Bögen zum Kassenhäuschen fahren, dort kurz warten und danach wieder per Lift zurück auf den Ausgangspunkt fahren. Der Fahrweg soll aus Geradenstücken und Kreisstücken bestehen.
Hinweis: die Formel für einen Kreis mit Radius r lautet:
 $K(t) = (r \cos(t), r \sin(t))^T$ für t aus dem Intervall $[0^\circ, 360^\circ]$
- (b) Durch Anklicken des Skifahrers startet die Animation. Ein Animationszyklus soll in 30 Sekunden durchlaufen werden.
- (c) Richten Sie den Skifahrer richtig entlang der Fahrtrichtung aus (sowohl auf dem geneigten Hang als auch auf den flachen Auslauf). Der Skifahrer soll einen Schlitten (einfach als Box modelliert) hinter sich herziehen, der allen Bewegungen des Skifahrers (zeitverzögert) folgt.

Aufgabe V.8 (zu Teil F aus der Vorlesung)

Texturierung ist eine häufig verwendete, wichtige Technik in der Computergraphik, um eine Szene attraktiv aussehen zu lassen, ohne die Anzahl der Polygone zu erhöhen.

- (a) Texturieren Sie das Kassenhäuschen und die Bäume.
- (b) Es soll ein Nachtskifahren geben: mit einem Schalter wird die Szene abgedunkelt. Der Skifahrer hält eine Fackel (modelliert als halbtransparenter roter Cone) in der Hand, der die Szene beleuchtet.

Aufgabe V.9 (Zusatzaufgabe)

Verschönern Sie die Szene! Fügen Sie 3D Sound oder eine Werbe-Video-Wand hinzu. Realisieren Sie eine Fahrt entlang Kurven. Usw.

Nutzen Sie zur Bearbeitung des Miniprojekts externe Informationsquellen (z.B. Bücher, Tutorials im Internet etc.) !!

