

VRML-Kurs Teil 3

Optische Eigenschaften von Objekten

Festlegung im `appearance`-Feld des `shape`-Knotens.

Zugehöriger `Appearance`-Knoten:

Appearance	material	SFNode
	texture	SFNode
	textureTransform	SFNode

Ins `material`-Feld wird ein `Material`-Knoten eingebunden:

Material	ambientIntensity	SFFloat	Bruchteil der Reflexion v. ungerichtetem Umgebungslicht
	diffuseColor	SFColor	Farbe unter Bestrahlung mit weißem Licht
	emissiveColor	SFColor	Farbe unabh. von der Beleuchtung
	specularColor	SFColor	Farbe der spiegelnden Reflexion
	shininess	SFFloat	Spiegelungsexponent
	transparency	SFFloat	Durchsichtigkeit

Ambiente Reflexion: diffuses Licht ohne bestimmte Richtung (indirekte oder Hintergrundbeleuchtung), abhängig nur von Anzahl der Lichtquellen, nicht von ihren Positionen. Der am-

biente Reflexionskoeffizient gibt an, welcher Anteil der ambienten Beleuchtung reflektiert wird (Farbe wie bei diffuser Reflexion).

Diffuse Reflexion: Licht wird von einer Fläche mit gleicher Intensität in alle Richtungen reflektiert, Helligkeit hängt ab von

- Winkel zwischen Flächennormale u. Richtung zur Lichtquelle
- Reflexionskoeffizienten für jede der 3 Grundfarben.

Spiegelnde Reflexion (specular reflection): Ausfallswinkel = Einfallswinkel (relativ zur Normalen).

Stärke des Intensitätsabfalls, wenn Objekt nicht exakt unter dem Reflexionswinkel betrachtet wird: *shininess* (Glanz)

- niedrige shininess: verschmierte, weiche Reflexe
- hohe shininess: kleine, harte Reflexe (glatte Oberfläche)

Transparenz: transparency-Koeffizient gibt an, welcher Anteil des auftreffenden Lichts die Fläche durchdringt

0 = undurchsichtig (default-Wert)

1 = total durchsichtig

Eigenleuchten (emissive color): inneres Glühen, wirkt aber nicht auf andere Objekte als Lichtquelle ein.

Licht- und Klangquellen:

Festlegung durch eigene Knoten (auf gleicher Ebene wie *Shape*)

PointLight (kugelsymmetrisch abstrahlende Punktlichtquelle, u.U. rechenintensiv)	location on intensity ambientIntensity attenuation	SFVec3f SFBool SFFloat SFFloat MFFloat	Ort d. Lichtqu. ein/aus Stärke (0...1) Beitrag zum ambienten Licht (0...1) 3 Koeff.: konst., lin., quadr. Abschwächung
--	---	--	---

	radius color	SFFloat SFColor	Reichweite Farbe
DirectionalLight (unendlich entfernte Punktlichtquelle, parallele Strahlen, geringste Rechenzeit)	direction on intensity ambientIntensity color	SFVec3f SFBool SFFloat SFFloat SFColor	Richtung der Lichtstrahlung ein/aus Stärke (0...1) Beitr. zum ambienten Licht (0...1) Farbe
SpotLight (Lichtkegel mit fester Begrenzung (cutOffAngle), größte Rechenzeit)	location direction on intensity ambientIntensity cutOffAngle beamWidth radius color	SFVec3f SFVec3f SFBool SFFloat SFFloat SFFloat SFFloat SFFloat SFColor	Ort d. Lichtqu. Richtung d. Kegelachse ein/aus Stärke (0...1) Beitr. zum ambienten Licht (0...1) halber Öffn.-winkel (Bogenmaß) Strahldicke Reichweite Farbe
Sound	direction intensity location maxBack maxFront minBack minFront priority source spatialize	SFVec3f SFFloat SFVec3f SFFloat SFFloat SFFloat SFFloat SFFloat SFNode SFBool	Richtung der Abstrahlung Lautstärke Ort der Quelle hinterer Rand der Hörbarkeits-Ellipse vord. Rand min. Reichweite Priorität AudioClip -Knoten Lokalisierbarkeit

AudioClip	description	SFString	Name des Audioclips
	loop	SFBool	Endlos-schleife
	pitch	SFFloat	Abspiel-geschwindigk. (Tonhöhe)
	startTime	SFTime	
	stopTime	SFTime	
	url	MFString	Adresse(n) der Audio-datei(en)
	duration_changed	SFTime) für Kommu-
	isActive	SFBool) nikation

Gerichtetes Licht (**DirectionalLight**) wirkt nur auf Knoten des eigenen Teilbaumes, dafür aber unabhängig von der Entfernung.

Beispiel:

3 blaue Kugeln sollen durch weißes Licht angestrahlt werden

- gerichtetes Licht
- Punktlicht
- Spotlicht

Wirkung auf nur je eine Kugel (durch Teilbaum-Zugehörigkeit bei der 1., bzw. durch Radius-Angabe bei der 2. und 3.)

```
#VRML V2.0 utf8
```

```
Transform
```

```
{
  translation -3 0 0
  children
  [
    Shape
    {
      geometry Sphere {}
      appearance Appearance
      {
        material Material
```

```

        { diffuseColor 0 0 1 }
    }
}
DirectionalLight
    { direction 0 -1 -1 }
]
}

PointLight
{
    location 0 3 3
    radius 3
}
Shape
{
    geometry Sphere {}
    appearance Appearance
    {
        material Material
        { diffuseColor 0 0 1 }
    }
}

SpotLight
{
    direction 0 -1 -1
    location 3 3 3
    radius 3
}
Transform
{
    translation 3 0 0
    children
        Shape
        {
            geometry Sphere {}
            appearance Appearance
            {
                material Material

```

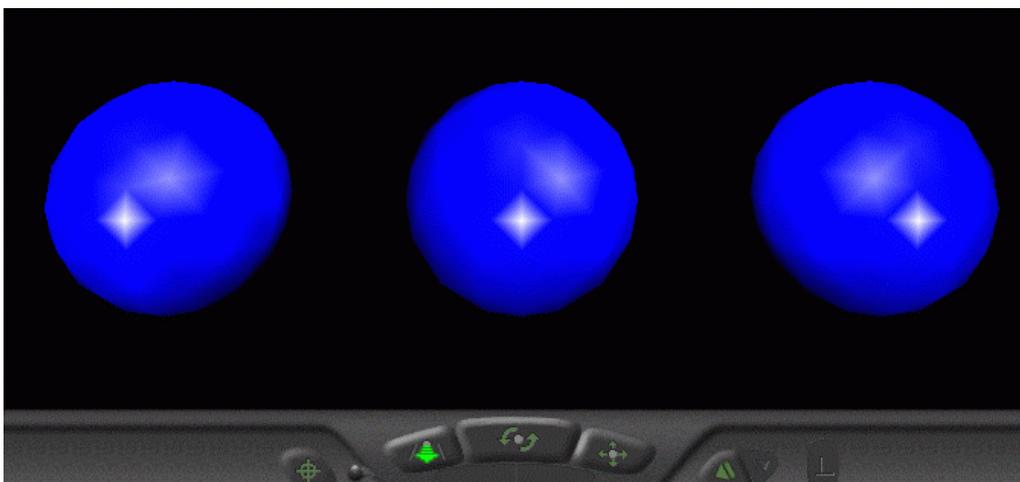
```
    { diffuseColor 0 0 1 }  
  }  
}
```

Ergebnis:



Modifikation, indem jeweils nach dem Feld
`diffuseColor 0 0 1`
unter den Materialeigenschaften noch eingefügt wird:
`specularColor 1 1 1`
`shininess 1`

Ergebnis:



Texturen

(schon in Teil 2 des Kurses benutzt)

- werden als Texturknoten unter dem Appearance-Knoten eingebunden
- 3 Typen von Texturen: Bitmap, Grafikdatei (URL; Formate JPG, GIF oder PNG), digitales Video (URL; Format MPEG1).

PixelTexture	image repeatS repeatT	SFImage SFBool SFBool	Bitmap horiz. Muster- Wiederholung vertikale Wdh.
ImageTexture	url repeatS repeatT	MFString SFBool SFBool	Adresse(n) zum Auffinden der Texturdatei horiz. Wdh. vertik. Wdh.
MovieTexture	loop speed startTime stopTime url repeatS repeatT duration_ changed isActive	SFBool SFFloat SFTIME SFTIME MFString SFBool SFBool SFTIME SFBool	Endlosschleife Abspiel- geschwindigkeit Adresse(n) zum Auffinden der Movie-Datei horiz. Wdh. vertik. Wdh.) dienen der) Kommunikation)

Texturen werden mit einem 2D-Koordinatensystem (s, t) versehen:



Blickpunkte (Kameras)

In einer Szene können beliebig viele Blickpunkte in Form von **viewpoint**-Knoten platziert werden ("Aussichtspunkte an interessanten Stellen")

– Browser verfügt über Schaltfläche, um Blickpunkte einer Szene nacheinander anzuspriegen bzw. auszuwählen

viewpoint	fieldOfView	SFFloat	Öffnungswinkel der Kamera
	orientation	SFRotation	Orientierung (Default: 0 0 1 0= Blickricht. neg. z-Achse)
	position	SFVec3f	Ort der Kamera
	jump	SFBool	
	description	SFString	Beschreibung
	set_bind	SFBool	eventIn-Feld
	bindTime	SFTime	eventOut-Feld
	isbound	SFBool	eventOut-Feld

Beispiel:

Def. zweier Blickpunkte von 2 Seiten (aus Richtung d. pos. und d. neg. z-Achse):

```

Viewpoint
{
  position 0 0 50
  orientation 0 0 1 0 # Voreinstellung
  description "Sicht von Norden"
}
Viewpoint
{
  position 0 0 -50
  orientation 0 1 0 3.14
  description "Sicht von Süden"
}

```

Mehrfachverwendung desselben Knotens (*object instancing*)

DEF und USE

- Dem Knoten muss mit **DEF** ein Name zugewiesen werden
- mit **USE** dann Referenz vermittelt dieses Namens
- die neuen Instanzen sind lediglich Zeiger (Referenzen)
- können transformiert, aber nicht mit neuen Attributen versehen werden

Beispiel:

```

#VRML V2.0 utf8
DirectionalLight {}
DEF Ball Shape # neuer Name
{
  geometry Sphere { radius 0.5 }
  appearance Appearance
  {
    material Material
    { diffuseColor 1 0 0 }
  }
}

```

Transform

```
{  
  scale 1.5 1.5 1.5  
  translation 2 0 0  
  children  
    [  
      USE Ball    # Referenz  
    ]  
}
```

Ergebnis:

