



Einführung in VRML Teil 3

Mehr Informationen zu Syntax, DEF/USE,
PROTO und Event-ROUTEs

Syntax in VRML-Files

- VRML unterscheidet Groß- und Kleinschreibung: **Sphere != sphere**
- Kommentare beginnen mit #
Alles **hinter #** in der Zeile ist Kommentar
- Whitespace sind *Space, Tab, Komma, Linefeed, Carriage Return*: Sie werden ausserhalb von Strings **ignoriert**.
- Namen dürfen keine Sonderzeichen oder Whitespace enthalten.

DEF und USE

- Mit DEF gibt man einem Knoten einen Namen, um ihn später in demselben VRML File wieder zu verwenden:

```
DEF mein_knoten Transform { ... }
```

- Mit USE kann man den Namen an anderer Stelle wieder benutzen, um einen gleichartigen Knoten einzufügen oder den Knoten zu referenzieren (z.B. in ROUTE Knoten, siehe weiter hinten):

```
ROUTE mein_knoten.collideTime TO SOUND2.startTime
```



Details zu PROTO

- Mit PROTO definiert man eigene Knoten.
- Eine PROTO Anweisung besteht aus Name, [Interface] und {Implementierung}.
- Der erste Knoten in der Implementierung definiert den “Typ” des neu definierten Knotens, weitere Knoten sind nicht referenzierbar und erscheinen auch nicht.
- USE / DEF funktioniert nicht über PROTO-“Grenzen” hinweg.



Events und ihr Einsatz (1)

- “Sensor” Knoten können Nachrichten absetzen, wenn Ereignisse eintreten.
- Verschiedene Knoten können derartige Nachrichten empfangen.
- StartTime und stopTime sind solche Nachrichten oder “Events”.
- Man kann z.B. einen Timer-Knoten definieren, der in bestimmten Intervallen Events erzeugt.

Events und ihr Einsatz (2)

- Eine Event-Verbindung definiert man durch ein ROUTE Statement.
- Damit werden ausgehende Events eines Knotens zu “Eingängen” eines anderen Knotens geleitet.
- Damit das geht, müssen die Knoten mit DEF benannt werden, z.B.:

```
DEF TIMER TimeSensor {  
    cycleInterval 5.0  
    loop TRUE  
}
```

Events und ihr Einsatz (3)

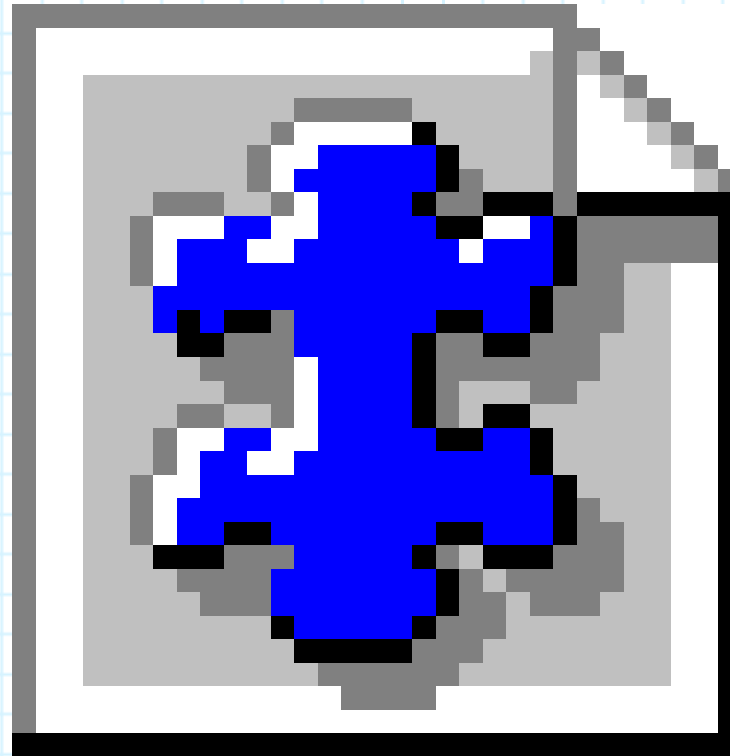
- Der Empfänger sieht z.B. so aus:

```
Sound {  
    minFront 10, minBack 10  
    maxFront 50, maxBack 50  
    source DEF SOUND AudioClip {  
        url "PICARD.WAV"  
    }  
}
```

- Dazu die Verbindung:

```
ROUTE TIMER.cycleTime TO SOUND.startTime
```

Event-Beispiel:



Timer im 5-Sekunden-Takt

Event-Beispiel: Proximity

