
Doppelstunde 20.Okt

3D-Grafikanimationen gestalten



Übersicht

Inhalte

- Wiederholung: Einfache und zusammengesetzte Objekte
- Diskussion: Bewertung des Kurses
- Praktische Übungen + Lösungen in Commsy stellen
- Exkurs: include Dateien
- Auswertung der Doppelstunde



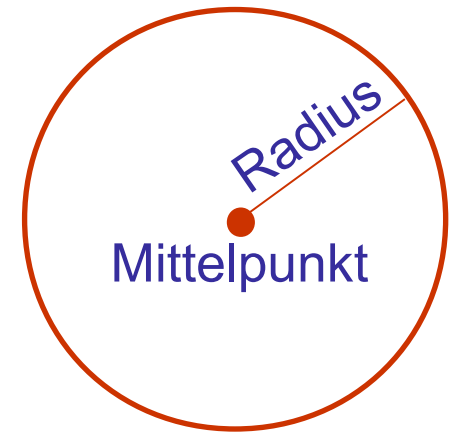
Wiederholung

Objektbeschreibungen haben immer den gleichen Aufbau.

Konzept

```
Objekt-Typ { spezielle Eigenschaften  
           allgemeine Eigenschaften  
}
```

- Typ-unabhängig:
 - Farbe, Oberflächenbeschaffenheit, Bewegung
- Typ-spezifisch:
 - Beispiel Kugel



Zusammengesetzte Objekte

merge

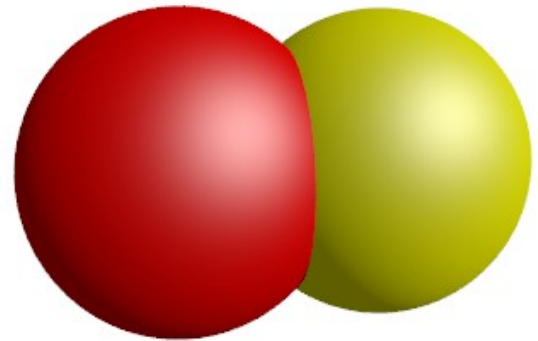
Die Verknüpfungen **merge** beschreibt eine Vereinigung zwischen zwei Objekten. Die beiden Objekte werden miteinander verschmolzen.

Konzept merge

```
merge { Beschreibung Objekt1  
      Beschreibung Objekt2  
}
```

Beispiel merge

```
merge { sphere { <0,0,0>, .8  
              pigment {color Red}  
            }  
      sphere { <1,0,0>, .8  
              pigment {color Yellow}  
            }  
}
```



Zusammengesetzte Objekte

difference

Die Verknüpfung **difference** beschreibt eine Differenz zwischen zwei Objekten.

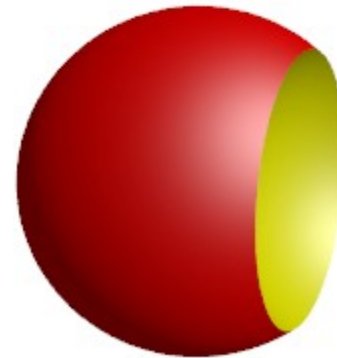
Teile des zweiten Objektes werden aus dem ersten Objekt herausgeschnitten.

Konzept

```
difference {      Beschreibung Objekt1
                Beschreibung Objekt2
            }
```

Beispiel

```
difference { sphere { <0,0,0>, .8
                    pigment {color Red}
                }
            sphere { <1,0,1>, .8
                    pigment {color Yellow}
                }
            }
```



Zusammengesetzte Objekte

intersection

Die Verknüpfung **intersection** beschreibt eine Überlappung zwischen zwei Objekten.

Der Teil, den beide Objekte gemeinsam haben, wird dargestellt.

Konzept

```
intersection { Beschreibung Objekt1  
                Beschreibung Objekt2  
}
```

Beispiel

```
intersection { sphere { <0,0,0>, .8  
                    pigment {color Red}  
                    }  
                sphere { <1,0,0>, .8  
                    pigment {color Yellow}  
                    }  
}
```



Allgemeiner Hinweis

Bei Fragen:

1. Schaut in den POV Ray guide
2. Helft Euch untereinander
3. Wenn alles nicht hilft, könnt ihr auch gerne mich fragen



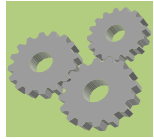
Dikussion: Bewertung des Kurses

Vorschläge:

1. Eigene Bilder oder Filme in Kleingruppen erstellen
2. Vorbereitete Übungen bearbeiten
3. Schriftlicher Test
 - Bis Ende dieser Doppelstunde kann jede Person entscheiden, welche Form bevorzugt wird:
 - Wer 1. oder 2. wählt, schreibt dies in die Diskussion im Commsy Raum (+ wer mit wem zusammenarbeitet)
 - Wer nichts weiter angibt, schreibt den schriftlichen Test



Praktische Übung:



1. Fertigstellen der bisherigen Übung
2. Erweiterung der Übung oder eigenes Bild gestalten

- Die bisherige Übung war: `kugelstuecke.pov`
- In dieser Datei sind 3 Kugeln. Verknüpft jeweils eine Kugel mit den Verknüpfungen `difference` und `intersection` mit einer `box` so, dass ihr eine Halbkugel, eine Dreiviertelkugel und eine Viertelkugel erhaltet.
- Erweiterung der Übung: Ergänze alle drei Kugelstücke um einen Hut

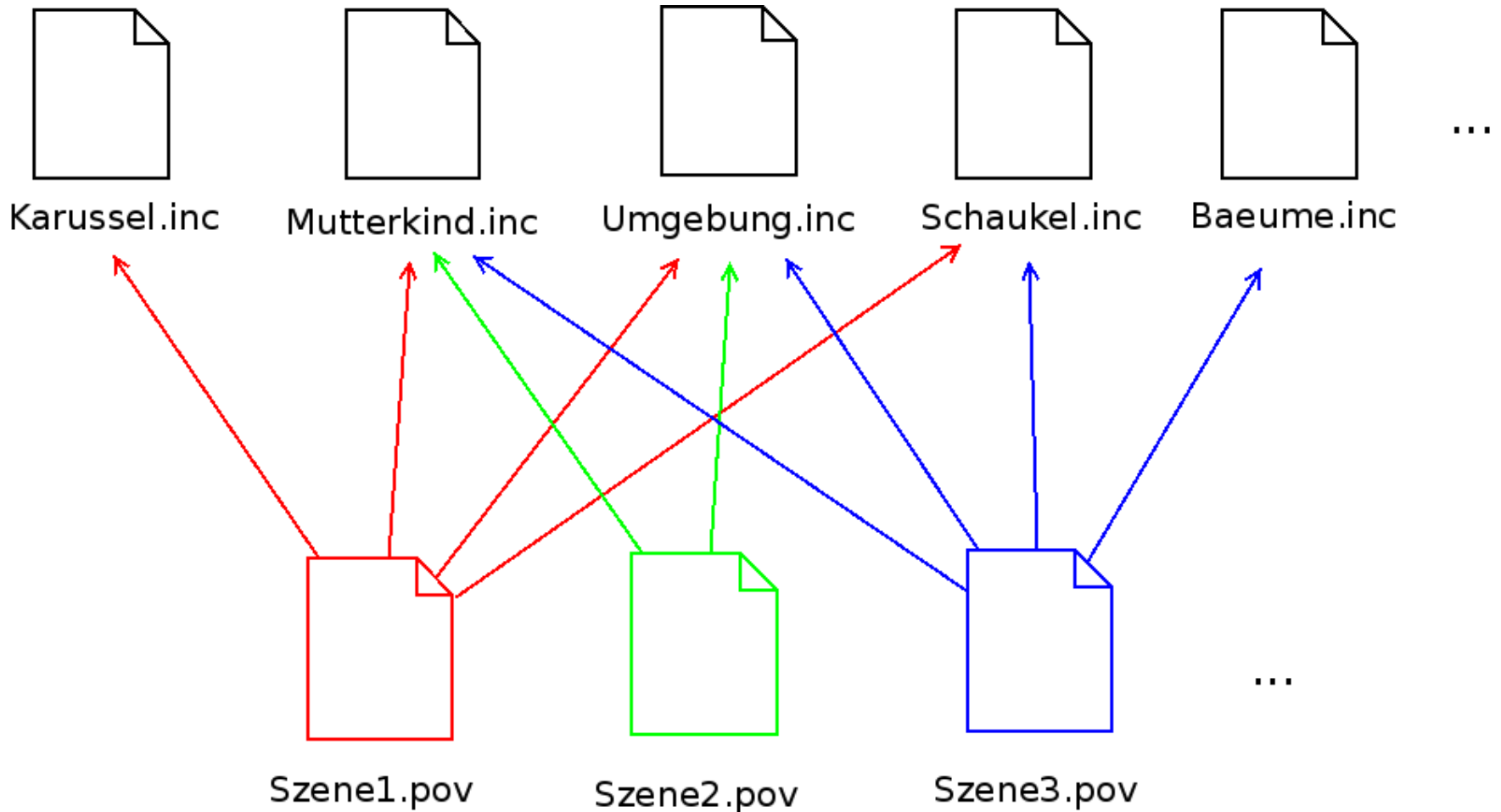


Exkurs: include-Dateien

- Größere Projekte werden meist in größeren Teams durchgeführt.
- Damit nicht alle an einer Datei arbeiten müssen, bieten alle besseren Programmiersprachen die Möglichkeit, den Programmcode auf mehrere Dateien aufzuteilen.
- In Povray geschieht dies mittels sogenannter *include-Dateien*.



Beispiel „Spielplatz-Film“



Include-Dateien

```
// include-Dateien mit Variablen:
#include "colors.inc"
#include „skies.inc“

// include-Dateien mit Objekten:
#include „koordinatenachsen.inc“
#include „saeulen.inc“

sky_sphere {
    S_Cloud5 // 'S_Cloud5' ist Variable aus skies.inc
}

light_source {
    <10, 20, -15>
    color White // 'White' ist Variable aus colors.inc
}

// Angezeigte Koordinatenachsen und Saeulen sind ebenfalls
// in den include-Dateien beschrieben
```

include-Dateien

Du kannst include-Dateien benutzen:

- Um deinen Code übersichtlich zu gliedern.
- Code-Teile wiederzuverwenden.
- Code-Stücke an andere weiterzugeben.
- Vorhandene include-Dateien benutzen um zu schauen, wie Farben, Texturen, Himmel und Ähnliches selber deklariert werden können.