

# ***Szenegraphen***

Ein Vortrag von Elvira Beck

# ***Was ist das?***

- Szenegraph ist hierarchische Szenererepräsentation, die Transformation nach unten vererbt
- Azyklischer Graph, oft sogar Baum
- Im CVK ist es aber nur ein Graph, dessen Nodes auch mehrere Eltern haben können



# ***Was sollte man vielleicht beachten?***

- Es gibt einen Wurzelknoten, unter dem die gesamte Szene aufgehängt wird
- Geht mit den Regeln einher, die von den gewählten Datenformaten o.Ä. auferlegt werden
- Es gibt Tools, die Szenegraphen erstellen können, aber händisch ist in unserem Fall eigtl. besser

# ***Wofür braucht man das?***

- Transformationsreihenfolge
- Bounding Volume Hierarchie
- State-Changes minimieren
- Speicher sparen
- Transparenzen korrekt zeichnen
- Trennung Darstellungsroutine und Geometrie
- Culling



# ***Nodes im Szenegraph im CVK***

- Group Nodes
- Transform Nodes
- Geometry Nodes
- Material Nodes
- Haben Kinder
- Haben Modelmatrix
- Haben Geometrie
- Haben Material

# ***Transformationsreihenfolge***

- Transformationsarten: Translation, Rotation, Skalierung
- Jeder Teilbaum kann in seinem eigenen lokalen Koordinatensystem verstanden werden
- Traversierung der Hierarchie akkumuliert die Matrizen der gesamten Welt und positioniert die lokalen Koordinatensysteme an den richtigen Stellen
- Wichtig, dass Zustand (hier Transformationen) gemerkt werden, sodass sie nach Verlassen des Teilbaums restauriert werden können



# ***Bounding Volume Hierarchie***

- Durch Szenegraph kann man auch direkt eine Bounding Volume Hierarchie aufbauen
- Für jeden Gruppenknoten wird auch Bounding Volume berechnet und abgespeichert
- BV ist dann meist in Weltkoordinaten und ist somit direkt für das Culling verwendbar (Traversierung nach oben)
- Wichtig, dass Szenenhierarchie räumliche Aufteilung aufweist
- Meist Bounding Box mit  $x_{min}/x_{max}$  und  $y_{min}/y_{max}$

# *Minimierung von State- Changes*

- OpenGL ist State-Machine
- Um effizienter zu sein, muss man State-Changes minimieren
- Geometrie reduzieren (Anzahl), einmal laden und mehrfach verweisen
- Materialänderungen minimieren
- Transparenzen zuletzt zeichnen
- Grafiksystem weiß, wo welches Objekt ist und deshalb ist das Aussuchen und Zeichnen einfacher



# *Culling*

- Nur das zeichnen, das gesehen wird
- Mit den BVs recht gut lösbar
- Man kann das Culling aber auch weiter fassen:
  - Räume und Verbindungen zw. diesen Räumen werden definiert
  - Nur aktueller Raum und sichtbare Räume werden gezeichnet (durch existierende Verbindungen)

## ***Sich bewegende Objekte***

- Wenn sich ein Objekt bewegt, muss man alle darüber liegenden Bounding Volumes aktualisieren
- Dynamische Objekte sollten von statischen getrennt werden
- Sinnvoll, alle dynamischen Objekte direkt unter die Wurzel zu packen



# ***Was hilft uns das jetzt?***

- Effizienter → schneller → besser
- Wir wollen schließlich in Echtzeit arbeiten
- Rekursives Rendering → man muss nicht alles händisch ändern
- Übersicht/ Ordnung/ Nachvollziehbarkeit
- Man kann interessante Dinge im Szenegrapg einarbeiten

# ***Was ist zusätzlich dadurch möglich?***

- Durch BV- Hierarchie ist Kollisionsabfrage effizient lösbar
- Man kann auch die Kamera im Szenegraph implementieren, sodass man mehrere Sichten verwenden kann, wenn man will
- Man kann auch mehrere Lichtquellen hinterlegen
  - Tag/Nacht- Wechsel
  - Lichtschalter
  - Etc.



## ***Andere interessante Ansätze***

- Man kann Objekte in unterschiedlichen Qualitätsstufen hinterlegen
  - z.B bei unterschiedliche Entfernungen
- Direkt Verhalten im Szenegraph speichern
  - Definition von Events als Auslöser (Bereiche, in die sich die Kamera bewegt, virtuelle Schalter ,...)
  - Wenn Auslöser betätigt wird, tritt Aktion ein
  - Erlaubt datengesteuerte Modellierung ganzer funktionstüchtiger Welten

# *Quellen*

- CG2 Vorlesung SS2014 Prof. Müller
- <http://wiki.delphigl.com/index.php/Szenengraph> 19.8.2014 10:50
- [http://www.informatik.uni-mainz.de/lehre/cg/SS2006\\_SCG/talks/SceneGraphs/scenegraphs.pdf](http://www.informatik.uni-mainz.de/lehre/cg/SS2006_SCG/talks/SceneGraphs/scenegraphs.pdf)

19.8.2014 11:45