

## Übung "Augmented Reality"

**Abgabetermin:** 18.12.06

### **Inhalt:**

*AR-Programmierung mit Java:* jARToolkit (für Java3D), Java3D (DirectX Renderer!), Java3D-VRML-Loader

### **Aufgabe 20 (P) jARToolkit, Java3D und VRML**

Laden Sie mit dem VRML-Loader für Java3D das Sheep-VRML-Model (Sheep.wrl). Modifizieren Sie für diese Aufgabe das mit im jarToolkit enthaltene Beispiel für Java3D so, dass statt dem Würfel das Schaf auf dem Hiro-Marker (Kantenlänge 6 cm) angezeigt wird.

### **Aufgabe 21 (P) Verdeckung virtueller durch reale Objekte**

Um AR-Szenen realistisch darzustellen, müssen sich reale und virtuelle Objekte korrekt verdecken. Dies wird durch das Z-Buffering unterstützt: Werden 3D-Modelle realer Objekte volltransparent gerendert, so wird nur der Z-Buffer verändert, nicht aber die Farbinformationen des aktuellen Frames.

Schneiden sie aus einem Blatt Papier ein 7 cm \* 7 cm großes Quadrat auf dem sich ein Hiro-Marker befindet (Kantenlänge 6cm). Rendern Sie nun einen 8 cm hohen Zylinder mit einem Durchmesser von 3cm, der das Papierquadrat genau in der Mitte so durchstößt, dass die halbe Höhe des Zylinders über dem Papier liegt. Der durch das Papierquadrat verdeckte Zylinderteil soll nicht mit dargestellt werden.

*Wichtige Java3D-Klassen:* OrderedGroup, TransparencyAttributes, RenderingAttributes

### **Aufgabe 22 (P) Einfacher Kollisionstest**

Verwenden Sie in dieser Aufgabe die Marker Kanji und Hiro (Seitenlänge je 6 cm). Auf dem Hiro-Marker soll ein Kegel angezeigt werden (Höhe 6cm, Durchmesser 6cm). Auf dem Kanji-Marker soll eine Kugel angezeigt werden (Durchmesser 6cm). Wird der Marker mit dem Kegel so an den Marker mit der Kugel herangeführt, dass die Kegelspitze die Kugeloberfläche durchstößt, soll die Kugel besonders gerendert werden (z.B. halbtransparent od. wireframe). Befindet sich die Kegelspitze außerhalb des Kugelvolumens, ist die Kugel normal zu rendern (solid).

*Wichtige Java3D-Klassen:* PolygonAttributes, RenderingAttributes

### **Aufgabe 23 (P) Animation**

Lassen Sie das Schaf zwischen zwei Markern (Hiro und Kanji) in einer Animation von einem Marker zum anderen „schweben“. Sie können dafür in Java3D beispielsweise den RotPosScaleTCBSplinePathInterpolator mit zwei entsprechenden Keyframes (tension, continuity und bias sollen 0 sein) verwenden. Eine Animation soll 4 Sekunden dauern und unendlich oft wiederholt werden.

**Aufgabe 24 (P) Multimarker-Unterstützung**

Unterstützung von Multimarkern ist zwar in der Schnittstelle des jARToolkit vorgesehen, wurde aber laut Dokumentation bisher nicht implementiert. Implementieren Sie deshalb eigene Funktionalitäten in Java, die es erlauben, mit Hilfe der vorhandenen jARToolkit-Funktionen für Single-Marker-Erkennung Multimarker-Unterstützung zu realisieren. Die Lagen der einzelnen Marker eines Multimarkers in Bezug zum Referenzpunkt können z.B. über Parameter einstellbar sein oder über eine Konfigurationsdatei (XML, VRML) geladen werden. Testen Sie die implementierten Funktionalitäten mit dem auf der AR-Webseite zur Verfügung gestellten Multimarker. Platzieren Sie ein beliebiges 3D-Objekt in der Mitte der zwischen den 4 Markern. Solange wenigstens einer der vier Marker erkannt wird, soll das Objekt an der festgelegten Position angezeigt werden.

*Hinweis:* Kann auch mit dem Standard-jARToolkit (nicht Java3D) gelöst werden

**Links:**

jARToolkit: <http://sourceforge.net/projects/jartoolkit/>

Java 3D: <https://java3d.dev.java.net/binary-builds.html>

Java3D-VRML-Loader: <https://j3d-vrml97.dev.java.net>

Java3D-Tutorial: <http://java.sun.com/developer/onlineTraining/java3d/>