



---

## Aufgabenblatt 5

13.07.2018

### Aufgabe 11: Kameramodell

**Hinweis:** Aufgabe gekürzt. Implementieren Sie eine virtuelle Kamera mit Look-at- und Projektionsmatrizen.

- a) Implementieren Sie eine Klasse zur Repräsentation einer virtuellen Kamera. Diese soll Member-Variablen und entsprechende get/set Methoden enthalten um alle relevanten Größen zur Spezifikation der Look-at- und perspektivischen Projektionsmatrix darzustellen, zusätzlich die Größe des Fensters in Pixeln (initial 400x400, Update kann über `CgWindowResizeEvent` erhalten werden) Alle Parameter der Kamera (in der Sichtweise *perspective*, siehe Folien) sollen per GUI einstellbar sein.
- b) Implementieren Sie direkt (d.h. ohne Nutzung der glm-Funktionalität) die look-at und Projektionsmatrix und übergeben Sie diese anstelle der Standard-Matrizen bei Änderung der Parameter an den Renderer (siehe `CgSceneControl::renderObjects()`).

### Aufgabe 12: 3D Picking Mechanismus

**Hinweis:** Aufgabe unverändert, Teil e) vereinfacht.

Implementieren Sie eine Funktionalität um Objekte zu selektieren. Dabei soll das das getroffene Dreieck und das zugehörige Objekt markiert werden.

- a) Implementieren Sie eine Funktionalität, die bei einem Rechtsklick in das Fenster (Maus-Events sind schon implementiert) den Schnittpunkt des Pick-Strahls mit den Objekten/Dreiecken der Szene berechnet.

Skalieren Sie dazu die Pixel-Koordinaten des Maus-Events so um, dass diese in x- und y-Richtung in Normalized Device Coordinates vorliegen und berechnen Sie die inverse Projektion eines Punktes auf der Near-Clipping-Plane und transferieren Sie diese Blickrichtung in Welt-Koordinaten (Trackball-Matrix nicht vergessen, siehe `CgSceneControl::renderObjects()`).

- b) Schreiben Sie eine Klasse zur Darstellung eines Strahls (z.B. `CgRay`) und implementieren Sie eine Funktionalität die die Schnittpunkte mit den Objekten ihres Szenegraphen berechnet, indem der Szenegraph den Strahl sukzessive in alle lokalen Koordinatensysteme seiner Objekte transferiert und den Schnittpunkt mit den Objekten berechnet.
- c) Markieren Sie die getroffenen Schnittpunkte (oder wahlweise nur den nächstgelegenen) mit je einer kleinen Kugel und geben sie als Ergebnis der Schnittpunktberechnung die Distanz zum nächstgelegenen Schnittpunkt zurück.

- d) Beschleunigen Sie ihre Schnittpunktberechnung, indem Sie eine Axis-Aligned-Bounding-Box in lokalen Koordinaten für jedes ihrer Objekte im Szenegraphen automatisch berechnen und zur Schnittpunkt-Berechnung nutzen. Die Anzeige der AABB (als Rechteck) sollen Sie per Gui ein- und ausschalten können, und zwar jeweils für das gerade selektierte Objekt.
- e) Erweitern Sie die Darstellung des Schnittpunkts, indem Sie die ID des getroffenen Dreiecks ausgeben.

**Hinweis:** Denken Sie an den kleinen Fehler im Skript, der hier relevant ist. Und zwar geht es um die Berechnung ob ein Strahl die AABB trifft. In Folie 5-202 wird (etwa bei 2/3 vom Code) folgende Abfrage gemacht: `if (t2>tmax) tmax=t2`. Diese muss korrekt `if (t2<=tmax) tmax=t2` heißen.