



Aufgabenblatt 2

09.04.2018

Aufgabe 4: Einfache Objekte - Zylinder und Kegel

Implementieren Sie jeweils eine Klasse - die direkt oder indirekt von `CgBaseTriangleMesh` ableitet - um einen Zylinder und einen Kegel zu realisieren. Die mathematische Definition des Objektes soll jeweils Teil der Klasse sein, ein Dreiecksnetz in variabler Auflösung soll auf Anforderung hin erzeugt werden. Folgende Teilaspekte sind zu beachten:

- Implementieren Sie eine Funktionalität mit der für einen Zylinder und Kegel ein Dreiecksnetz in variabler Auflösung erzeugt werden kann. Dazu soll sowohl die Auflösung (d.h. Anzahl der Segmente) in Rotationsrichtung als auch in der Höhe frei gewählt werden können. Ergebnis soll jeweils ein geschlossenes Netz sein. Die Lage in lokalen Koordinaten können Sie so wählen, dass die Berechnung der Punkte möglichst einfach erfolgen kann.
- Machen Sie diese Parameter mit GUI-Elementen steuerbar. Implementieren Sie also analog zu der Farbwahl bei der vorherigen Aufgabe entsprechende GUI-Elemente und eigene Events um ihren Controller zu benachrichtigen. Das jeweilige Objekt soll dann seine Auflösung entsprechend ändern. Eine entsprechende Reset-Funktionalität die den Speicher wieder frei gibt und neu anlegt bei veränderten Parametern versteht sich von selbst.
- Berechnen Sie für jeden Vertex die geschätzte Oberflächennormale. Dabei sollen Sie die spezielle rotatorische Struktur der Objekte ausnutzen und eine Routine erstellen die die Normalen möglichst elegant und genau berechnet.
- Zeichnen Sie die Normalen analog zum Würfel als Linienstück an den jeweiligen Punkt.
- Die Funktionalität aus Teil d) soll per GUI ein- und ausgeschaltet werden können. Die Linien-Objekte sollen also jeweils automatisch erstellt und wieder gelöscht werden. Dazu ist es notwendig - da Sie viele Objekte erzeugen - dass Sie einen Mechanismus implementieren der für ein beliebiges Objekt automatisch eine eindeutige ID vergibt. Dieser Mechanismus soll für alle Objekte ihres Systems verwendbar sein. Recherchieren Sie ein passendes Design-Pattern (= Software-Entwurfs-Muster) und implementieren Sie entsprechend.

Aufgabe 5: Einfache Objekte - Rotationsobjekte mit Unterteilungskurven

Implementieren Sie eine Klasse - die direkt oder indirekt von `CgBaseTriangleMesh` ableitet - um einen Rotationskörper zu realisieren, der als Kontur eine Unterteilungskurve besitzt. Folgende Teilaspekte sind zu beachten:

- a) Implementieren Sie den Lane-Riesenfeld Unterteilungsalgorithmus. Machen Sie diesen per GUI steuerbar indem Sie die Anzahl der Mittelungsschritte wählbar machen sowie einen Button der einen Unterteilungsschritt ausführt (wie immer entsprechende GUI-Elemente und Events implementieren). Der Speicherverbrauch für den Algorithmus so klein wie möglich sein. Die Verwaltung des Speichers soll also wie in der Vorlesung gezeigt ohne zusätzliche Listen auskommen. Zeichnen Sie die Kurve zunächst als PolyLine mit der Klasse, die Sie auch für die Normalen verwendet haben (falls nicht sowieso schon vorhanden: beliebige Punkt-Zahl pro Polyline zulassen)
- b) Implementieren Sie eine Reset-Funktionalität für ihre Kurve und einen entsprechenden Button im GUI dazu.
- c) Erstellen Sie einen Rotationskörper (Anzahl rotatorische Segmente per GUI wählbar, Anzahl Segmente in der Höhe entsprechend der Punkte-Anzahl der Kurve), der ihre Unterteilungskurve um die lokale y-Achse rotiert. Falls die Achse durch die beiden Endpunkte der Kurve geht, soll ein geschlossenes Netz erzeugt werden, falls nicht ein offenes Netz (beide Varianten sollen vorgeführt werden können). Die Erstellung des Rotationskörpers soll per Button-Klick gestartet werden.
- d) Berechnen Sie auch für dieses Objekt möglichst effizient und elegant die Normalen pro Vertex und zeigen Sie diese wahlweise als Linien-Segmente an (analog zur vorherigen Aufgabe ein- und ausschaltbar). Beachten Sie auch hier die spezielle Struktur des Objekts.
- e) Erweitern Sie die Steuerungsmöglichkeit der Objektauflösung wie folgt: Für eine gegebene Unterteilungskurve soll unabhängig von deren Punkte-Zahl neben der rotatorischen Auflösung auch eine Auflösung in der Höhe wählbar und veränderbar sein (GUI-Funktionalität). Berechnen Sie dann die Mindest-Anzahl an notwendigen Unterteilungen um die gewünschte Auflösung (oder minimal mehr) zu erreichen, führen Sie diese aus und erstellen Sie den Rotationskörper entsprechend. Auch hier ist eine saubere Reset-Funktionalität für ihr Objekt notwendig.