HOCHSCHULE
HANNOVER
UNIVERSITY OF
APPLIED SCIENCES
AND ARTS
–
Fakultät IV
Wirtschaft und
Informatik

Übungen zur Vorlesung Computergrafik 1

Soommersemester 2018

Prof. Dr. Ingo Ginkel

Aufgabenblatt 3

16.04.2018

Aufgabe 6: Vervollständigung Mesh-Funktionalität

Für diese Aufgabe steht ihnen eine Funktionalität zum Laden von .obj-Dateien zur Verfügung.

a) Kopieren Sie dazu die beiden Dateien aus `CgUtils.zip` in ihr Projekt z.B. in einem neuen Unterverzeichnis `CgUtils`. Fügen Sie die Dateien zu ihrem Projekt hinzu, so dass diese mit-compiliert werden.

b) Zur Ansteuerung der Loader-Funktionalität implementieren Sie ein GUI-Element um eine Datei auszuwählen. Ein entsprechender Menü-Eintrag und ein `slot` der bei Auswahl aktiviert wird ist bereits in `CgQtGui` vorhanden. Ergänzen sie dort z.B.

`QString file =QFileDialog::getOpenFilename(...).`

Übergeben Sie den Dateinamen als `std::string` an eine eigenes Event, das den Datei-Namen über das Observer Pattern an `CgSceneControl` schickt.

In der `HandleEvent`-Methode von `CgSceneControl` können Sie dann den obj-Loader benutzen (ein paar Beispiel-Dateien finden Sie in `CgData.zip`):

```
std::string filename = "somename.obj";
ObjLoader loader;
loader.load(filename);

std::vector<glm::vec3> pos;
loader.getPositionData(pos);

std::vector<unsigned int> indx;
loader.getFaceIndexData(indx);

// hier jetzt das Dreiecksnetz initialisieren
// und beim Renderer anmelden

m_renderer->redraw();
```

c) Das Objekt soll später einmal so dargestellt werden, dass die Kanten in der Geometrie durch geschickte Beleuchtung und Schattierung verwischt werden und das Objekt so glatter erscheint als es eigentlich ist. Dazu ist es notwendig eine Normale nicht mehr pro Dreieck, sondern pro Punkt anzugeben. Diese Punktnormale wird als Mittelwert der Normalen aller Dreiecke berechnet, in denen der jeweils betrachtete Punkt vorkommt.

Implementieren Sie einen Algorithmus der dies in Zeitkomplexität $O(n)$ realisiert (n steht hier gleichermaßen für die Anzahl der Punkte oder Anzahl der Dreiecke) , falls immer alle Punktnormalen auf einmal berechnet werden sollen.

Hinweis: Der obj-Loader ist in der Lage analog zu den Punkten auch in der Datei gespeicherte Normalen einzulesen. Diese Funktionalität soll zunächst explizit nicht benutzt werden, sondern die Normalen wie oben beschrieben berechnet werden (die Normalen-Info ist auch nicht immer in jeder obj-Datei vorhanden).

Aufgabe 7: Transformationen: Skalierung, Verschiebung, Rotationen

In dieser Aufgabe sollen Transformationen in homogenen Koordinaten für ein einzelnes Objekt realisiert werden. Es ist nicht nötig die Transformation selbst auf die Punkte anzuwenden, sondern es soll die jeweils anzuwendende Transformationsmatrix berechnet werden, der Renderer führt die Transformation dann aus. Diese Transformationsmatrix wird dann in `CgSceneControl::renderObjects()` beim Render-Aufruf übergeben. Momentan ist das eine Einheitsmatrix, diese sollen Sie entsprechend ersetzen. Generell sind diese Objekttransformationen eine Funktionalität die vom `Control` übernommen und dort implementiert wird, das Objekt in lokalen Koordinaten wird dabei gar nicht verändert.

a) Implementieren Sie eine Skalierungs-Funktionalität, diese soll über Tastendruck (z.B. mit +/-) gesteuert werden, Key-Events sind für das Observer-Pattern schon implementiert und können in der `HandleEvent()` - Methode direkt abgefragt werden. Die Skalierung soll in jeder Koordinaten-Richtung gleich sein, aber prinzipiell soll beliebige Skalierung in jeder Koordinatenrichtung vorgesehen werden (Argument ist also ein vec3). Erstellen Sie die entsprechende Matrix und übergeben Sie diese beim render-Aufruf.

b) Analog soll per Tastendruck (z.B. x,y,z- Taste) eine Rotation um die lokale x-, y- oder z-Achse des Objekts ausgeführt werden können. Um diese Funktionalität optisch besser einschätzen zu können zeichnen Sie die Koordinatenachsen als 3D- Objekte in 3 verschieden Farben mit Hilfe der Funktionalität für Zylinder und Kegel aus der vorherigen Aufgabe.

c) Ergänzen Sie eine Funktionalität zur Verschiebung, der Verschiebungsvektor soll dabei über das GUI gewählt werden, die Verschiebung über ein entsprechendes Event über das Observer-Pattern ausgelöst werden.

d) Erweitern Sie ihre Implementierung derart, dass die Kombination von Transformationen korrekt abgearbeitet wird. Dazu ist es nötig z.B. das „Wandern“ beim Skalieren eines Objektes durch eine entsprechende Translation zu vermeiden etc...