Computergrafik 1

Lab 7

Dieses Lab ist eine Weiterführung von Lab 6.

Aufgabe 1 Normalen selber berechnen

(a) Implementieren Sie in der Datei Vec3.js die Methoden `dot`, `euclideanLength`, `add`, `sub`, `cross`, `scale` und `normalize` gemäß der Dokumentation im Code.

(b) Prüfen Sie in `TriangleMeshGL`, ob das ein reinkommende `simpleMeshIO` Normalen besitzt. Sollten keine vorhanden sein, stellen Sie sicher, dass zunächst kein WebGL Array Buffer erzeugt und gebunden wird.

(c) In der `setup` Methode von Mesh3DApp können Sie die Zeile

```
const streamReader = await loadBinaryDataStreamFromURL("./data/bunny.smm")
```

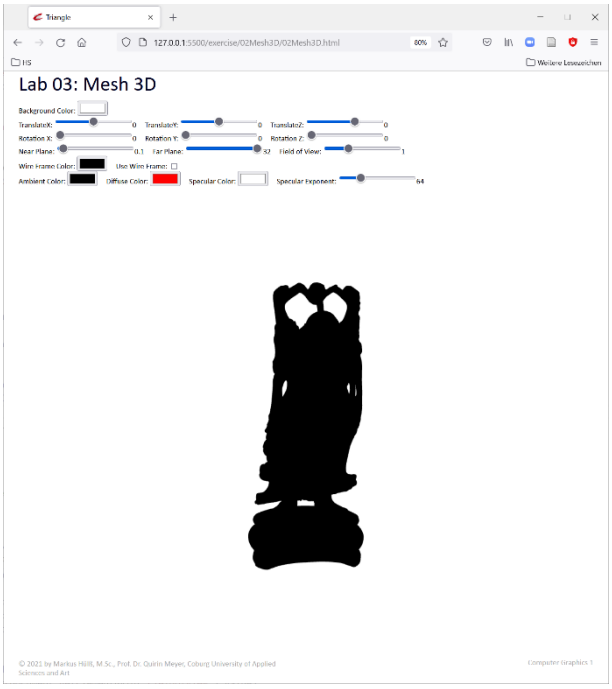
durch

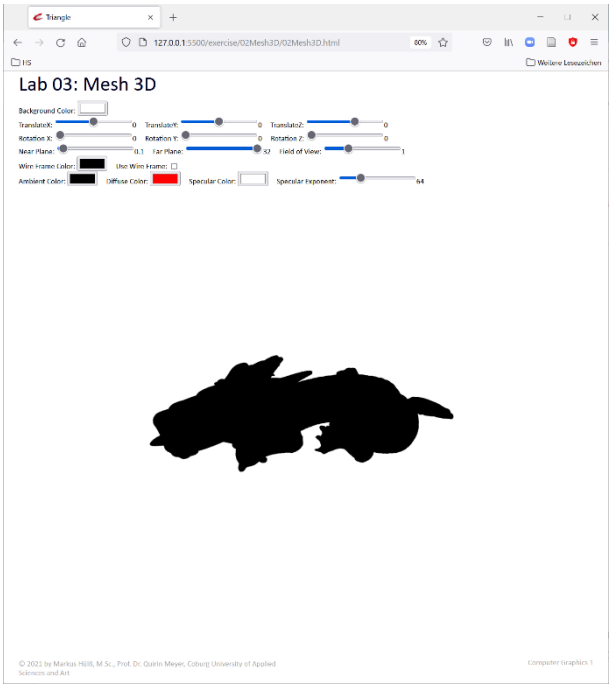
```
const streamReader = await loadBinaryDataStreamFromURL("./data/happy.smm")
```

oder

```
const streamReader = await loadBinaryDataStreamFromURL("./data/dragon.smm")
```

ersetzen. Stellen Sie aber vorher sicher, dass die Dateien dragon und happy auch im Data Ordner liegen. Beim Zeichen sollten Sie folgende Bilder erhalten:





(d) In (b) haben sie ja bereits in `TriangleMeshGL` geprüft, ob das reinkommende `simpleMeshIO` Normalen besitzt. Tut es dies nicht so müssen Sie die Normalen selber bestimmen! Legen Sie dazu die Methode

```
/**
Computes the normals of a vertex by averaging its neighboring face normals.
Each face normal is weighted by the face area.
@param {number[]} triangles Index buffer contained triangles.
@param {number[]} positions 3D Positions.
*/
computeAverageNormals(triangles, positions)
```

an und rufen Sie diese im Konstruktor auf!

Der Algorithmus zur Berechnung sieht wie folgt aus:

1. Zunächst wird ein Vertex-Normalen Array angelegt, das für jeden Vertex eine Normale bereitstellt. Initial sind diese Normale alle mit $[0; 0; 0]^T$.
2. Für jedes Dreieck…
   a. … werden die drei Vertexindizes gelesen,
   b. … werden mittels der Vertexindizes die drei Vertexposition ermittelt,
   c. … werden aus den drei Vertexposition wird die Dreiecksnormale berechnet,
   d. … wird die Dreiecksnormal auf alle drei beteiligten Vertex-Normalen addiert, die an dem Dreieck liegen.
3. Abschließend wird jede Vertex-Normal normalisiert

Implementieren Sie diesen Algorithmus. Sie sollten dann folgende Ausgabe erhalten: