

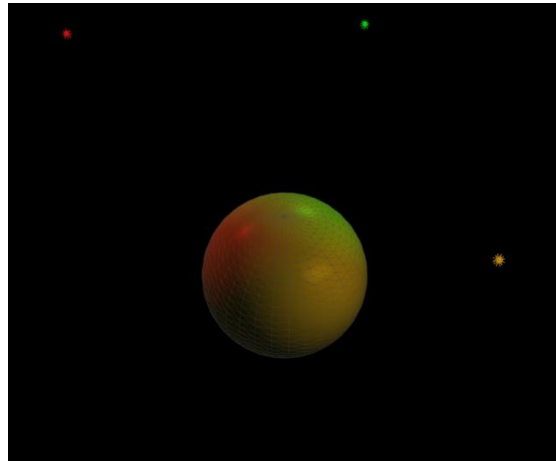


Grundlagen der Computergrafik — Praktikum

Beleuchtung

Ziel

Am Ende dieses Praktikumsteils haben Sie die Beleuchtungsberechnung mit dem Phong-Beleuchtungsmodell selbst implementiert:



Vorbereitung

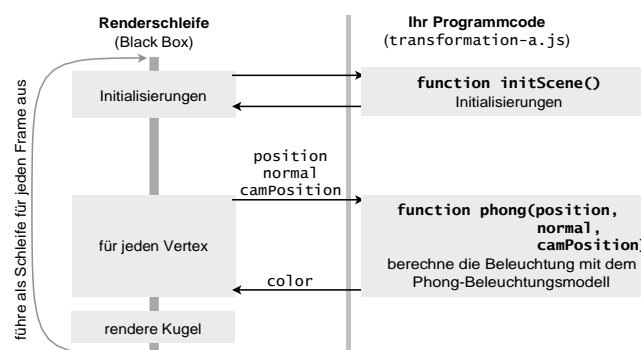
Wiederholen Sie die Inhalte aus dem Vorlesungsabschnitt *Beleuchtung*.

Zusammenspiel von Black Box und Ihrem Code

In diesem Praktikumsteil beinhaltet die Black Box ebenfalls eine Rendschleife. Während der Initialisierung können Sie in der Funktion `initScene()` Lichter anlegen und bei der Blackbox durch Aufruf der Funktion `registerLights(...)` anmelden. Nach dem Aufruf dieser Funktion werden Licht-Icons durch die Black Box gerendert. Diese befinden sich an der jeweiligen Lichtposition und helfen Ihnen, die Beleuchtung zu beurteilen.

Während des Renderings ruft die Rendschleife der Black Box für jeden einzelnen Vertex Ihre Funktion `phong(...)` auf. Die Position des Vertex (`position`), die Normale am Vertex (`normal`) sowie die Kameraposition (`camPosition`) werden als Parameter übergeben.

Einen Überblick über diesen Ablauf gibt die folgende Abbildung.



Nützliches zur Programmierung

Datenaustausch

Wie Sie auf die Attribute eines Objekts zugreifen können, zeigt folgendes Beispiel:

```
let colour = {"r" : 0.9,  
             "g" : 0.6,  
             "b" : 0.3};
```

Das Objekt `colour` mit den Attributen `r`, `g` und `b` wird angelegt und den Attributen Werte zugewiesen. Danach liefert `colour.r` den Wert des Attributs `r`. Um den Wert des Attributs zu ändern, verwenden Sie beispielsweise:

```
colour.r = 0.5;
```

Vektoren und Rechenoperationen

Während der Beleuchtungsberechnung werden 3D Spaltenvektoren verwendet. Die Black Box in Ihrer Projektmappe stellt Ihnen 3D Spaltenvektoren zur Verfügung. Mit folgendem JavaScript-Code legen Sie einen Nullvektor (alle Elemente mit Wert Null) an:

```
let nullvektor = new THREE.Vector3();
```

In diesem Praktikumsteil werden Sie die Differenz zweier Vektoren `vecA` und `vecB` benötigen und in einem dritten Vektor `vecC` speichern müssen:

```
vecC.subVectors(vecA, vecB);
```

Mit folgendem Code können Sie einen Vektor normalisieren:

```
vecA.normalize();
```

Wichtig: Der Vektor `vecA` wird dabei verändert und hat anschließend die Länge 1.

Das Skalarprodukt aus zwei Vektoren können Sie mit folgendem Code berechnen und in einer Variablen `f` speichern:

```
let f = vecA.dot(vecB);
```

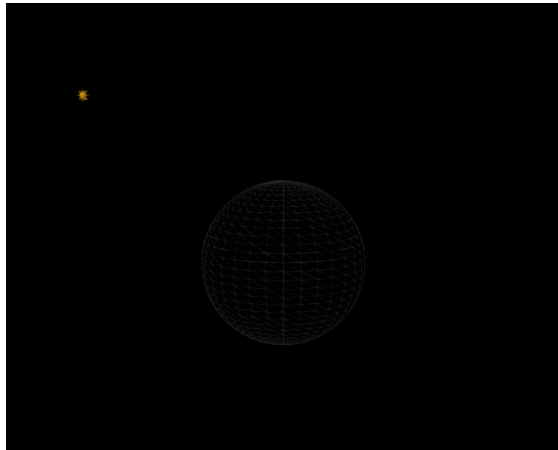
Um einen Vektor mit einem Skalar zu multiplizieren, können Sie folgenden Code nutzen:

```
vecA.multiplyScalar(scalar);
```

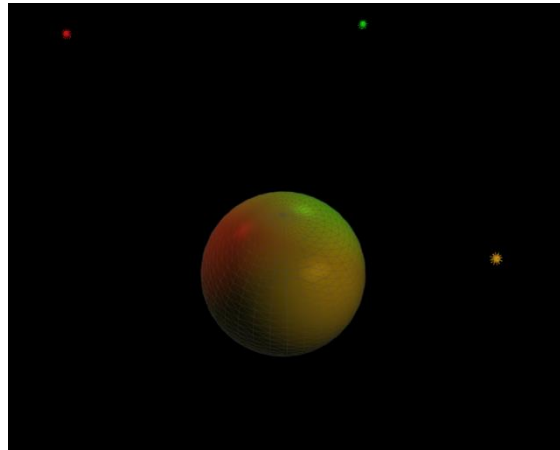
Aufgaben

Öffnen Sie die Datei `illumination.html` in Ihrem Browser. Sie sollten auf der Html-Seite einen Rahmen sehen, in dem sich eine Kugel im Koordinatenursprung befindet. Ein Icon für ein Punktlicht wird ebenfalls angezeigt. Die Beleuchtung wird noch nicht berechnet. Zur Orientierung wird zusätzlich das Drahtgittermodell der Kugel auf ihrer Oberfläche angezeigt. In der Datei `illumination.js` ist bereits das ambiente Licht der Szene sowie ein Punktlicht angelegt. Die verwendeten bzw. benötigten Attribute entnehmen Sie dem Quellcode.

zu Beginn



am Ende dieses Praktikumsteils



1. Ambienter Lichtanteil

Implementieren Sie in der Funktion `phong(...)` in der Datei `illumination.js` die Berechnung des ambienten Anteils der Beleuchtung nach dem Phong-Beleuchtungsmodell. Legen Sie global im Skript den Reflexionskoeffizienten für den ambienten Lichtanteil an. Beurteilen Sie das Bildergebnis und beheben Sie evtl. auftretende Fehler.

2. Diffuser Lichtanteil

Implementieren Sie in der Funktion `phong(...)` in der Datei `illumination.js` die Berechnung des diffusen Anteils der Beleuchtung nach dem Phong-Beleuchtungsmodell für das vorhandene Punktlicht. Legen Sie global im Skript den Reflexionskoeffizienten für den diffusen Lichtanteil an. Beurteilen Sie das Bildergebnis und beheben Sie evtl. auftretende Fehler.

3. Spekularer Lichtanteil

Implementieren Sie in der Funktion `phong(...)` in der Datei `illumination.js` die Berechnung des spekularen Anteils der Beleuchtung nach dem Phong-Beleuchtungsmodell für das vorhandene Punktlicht. Legen Sie global im Skript den Reflexionskoeffizienten für den spekularen Lichtanteil sowie ggf. weitere benötigte Materialeigenschaften an. Beurteilen Sie das Bildergebnis und beheben Sie evtl. auftretende Fehler.

4. Weitere Punktlichter

Legen Sie mindestens zwei zusätzliche Punktlichter an. Passen Sie die Beleuchtungsberechnung so an, dass eine beliebige Anzahl von Punktlichtern korrekt berücksichtigt werden kann. Schaffen Sie eine visuell ansprechende Lichtstimmung. Beurteilen Sie das Bildergebnis und beheben Sie evtl. auftretende Fehler.

Bearbeitung, Abgabe, Bewertung

Die Abgabe erfolgt gruppenweise. Ein Gruppenmitglied lädt dazu die Datei `illumination.js` auf der Moodle-Seite des Kurses als komprimierte Zip-Datei hoch.

Wichtig: Tragen Sie unbedingt Ihre Gruppe, den Studiengang sowie die einzelnen Gruppenmitglieder im führenden Kommentarblock der Datei ein. Benennen Sie die Datei folgendermaßen:

STUDIENGANG-GRUPPE-NAME1-NAME2-transformation.zip (Beispiel: BMT-A-MayerRichter-transformation.zip)

Punkteverteilung (10 Punkte)

Für diesen Praktikumsteil erhalten Sie maximal 10 Punkte. Verspätete Abgaben werden nicht berücksichtigt. Bei Feststellung eines Täuschungsversuchs erhalten die betroffenen Gruppen keine Punkte für das Praktikum.

Aufgaben 1 - 4 jeweils:	2 Punkte
Aussagekräftige Kommentierung & Variablen:	1 Punkt
Performance:	1 Punkt