

Protokol Übung 02 - der Raytracer

Johann Roehl, Bodo Wissemann, Phillip Zuschlag

1 Zum Inhalt dieses Protokolls

Das Protokoll behandelt die Implementierung des Raytracers mit allen dafür erforderliche Klassen:

- Camera (abstract)
- OrthographicCamera
- PerspectiveCamera
- Color
- Geometry (abstract)
- Plane
- Sphere
- Triangle
- AxisAlignedBox
- Hit
- Ray
- World

2 Die Kamera-Klassen

Zu erst wurde die abstrakte Klasse Camera implementiert und anschließend die beiden Klassen OrthographicCamera und PerspectiveCamera, welche von Camera erben. Die in den jeweiligen Konstruktoren und Methoden umzusetzenden mathematischen Formeln wurden Schritt für Schritt über die in der Matrizen und Vektoren-Bibliothek implementierten Methoden implementiert. Beispiel hierfür ist etwa die Methode rayFor() der Klasse OrthographicCamera mit den Zwischenergebnissen doub1 und doub2. Nach der Implementierung aller Klassen und einem ersten Rendern von Bildern mussten wir noch einige wenige Fehler korrigieren. Zum Beispiel war \vec{u} nicht normalisiert und alle Bilder erschienen zu groß.

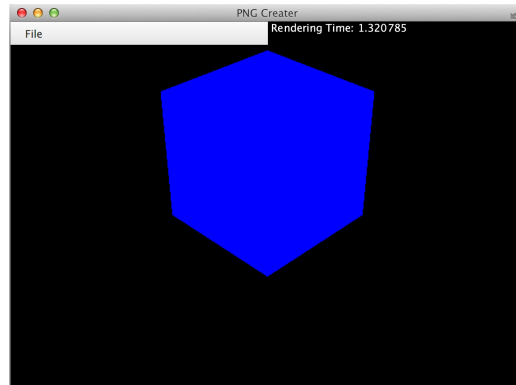


Abbildung 1: Bild bei nicht normalisiertem \vec{u}

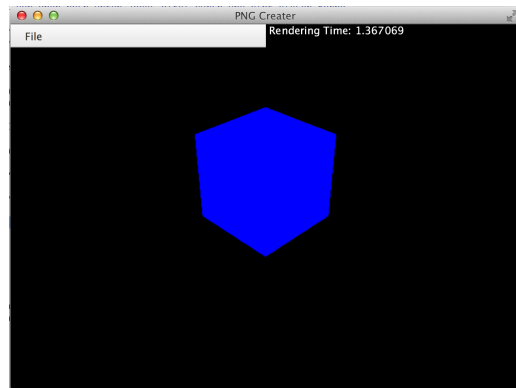


Abbildung 2: Bild bei normalisiertem \vec{u}

3 Die Geometrien

Die abstrakte Klasse geometry wurde erstellt und anschließend die Klassen Plane, Sphere, Triangle und AxisAlignedBox, welche von geometry erben. Problematisch war, dass ein Test der Klassen erst nach vollständiger Implementierung aller Komponenten des Raytracers möglich war - auch hier war somit nach Ende der Implementierung noch eine Fehlersuche nötig. Beispielsweise war eine Formel in der Klasse Plane noch fehlerhaft.

4 Die Klassen Ray, Color und World

Die Klassen Ray, Color und World bereiteten keine Probleme.

5 Weitere Schwierigkeiten

Unser Bild stand - wie bereits in der Vorlesung erwähnt wurde - zunächst auf dem Kopf. Wir versuchten zunächst, den Y-Wert in den Klassen zu drehen. Da der Wert final ist kann man sich ohne weiteres nicht verändern. Wir versuchten zunächst, Methoden zur Erstellung invertierter Objekte der jeweiligen Klassen zu erstellen und zu nutzen. Erst später kamen wir auf die Idee, die Umrechnung in der geschachtelten Vor-Schleife der Raytracer-Klasse vorzunehmen. Statt der y-Wert wird über die Formel $height - 1 - y$ umgerechnet.