

Computer Grafik Bericht - Übung 1

Gruppe Mineral: Lena Meßmer, Christian Mehns, Sophie Wirth

1. Dezember 2014

Inhaltsverzeichnis

1	Einleitung	1
2	Strahl	2
2.1	Ray	2
3	Camera	2
3.1	Camera (abstrakte Klasse)	2
3.2	OrthographicCamera	2
3.3	PerspectiveCamera	2
4	Farbe	3
4.1	Color	3
5	Geometrie	3
5.1	Geometry (abstrakte Klasse)	3
5.2	Hit	4
5.3	Plane	4
5.4	Sphere	4
6	World	4
6.1	World	4
7	RayTracer	5
7.1	RayTracer	5
7.2	TestRay	5

1 Einleitung

Das ist der zweite Bericht der Gruppe in CG, bestehend aus Lena Meßmer, Christian Mehns und Sophie Wirth.

2 Strahl

2.1 Ray

Bearbeitungszeit: 30 min

Die Klasse repräsentiert einen Strahl.

Der Konstruktor nimmt als Parameter den Ursprung und die Richtung des Strahls entgegen und speichert sie als private Klassenattribute.

Die Methode `at` gibt den Punkt an einer gegebenen Distanz, auf dem Strahl, zurück.

Die Methode `tOf` gibt die Distanz als `double` zurück, die ein Punkt auf dem Strahl vom Punkt-Ursprung entfernt ist.

3 Camera

3.1 Camera (abstrakte Klasse)

Bearbeitungszeit: 1 h

Im Konstruktor werden die Kameraposition (eye position) als `Point3` übergeben, sowie der gaze- und der up-Vektor.

Außerdem werden die Vektoren `u,v,w` berechnet, welche als öffentliche Klassenattribute gespeichert werden.

Die Methode `rayFor` liefert für einen bestimmten Pixel einen Strahl zurück.

Als Parameter werden der Methode Breite und Höhe des Bildes übergeben, sowie die Koordinaten des Pixels, für den der Strahl generiert werden soll.

3.2 OrthographicCamera

Bearbeitungszeit: 40 min

Stellt eine orthographische Kamera dar und ist eine Unterklasse von `Camera`. Im Konstruktor wird, neben Kamera-Position, gaze- und up-Vektor, auch ein Parameter für einen Skalierungsfaktor übergeben, dieser bestimmt den Sichtbereich der Kamera. Die Methode `rayFor` wird überschrieben. Um einen neuen Strahl zurückzuliefern, berechnet die Methode die Werte `Origin` und `Direction` wie folgt:

$$direction = -\vec{w} \quad (1)$$

$$origin = \vec{e} + s\left(\frac{x - \frac{w-1}{2}}{w-1}\right) \cdot \vec{u} + s\left(\frac{y - \frac{h-1}{2}}{h-1}\right) \cdot \vec{v} \quad (2)$$

3.3 PerspectiveCamera

Bearbeitungszeit: 40 min

Stellt eine perspektivische Kamera dar und ist eine Unterklasse von `Camera`. Im Konstruktor wird, neben Kamera-Position, gaze- und up-Vektor, auch ein Parameter für

den Öffnungswinkel übergeben. Die Methode `rayFor` wird überschrieben. Um einen neuen Strahl zurückzuliefern, berechnet die Methode die Werte `Origin` und `Direction` wie folgt:

$$origin = \vec{e} \quad (3)$$

$$direction = \frac{\vec{r}}{|\vec{r}|} \quad (4)$$

wobei

$$r = -w \cdot \frac{(\frac{h}{2})}{\tan \alpha} + u \cdot (x - (\frac{w-1}{2})) + v \cdot (y - (\frac{h-1}{2})) \quad (5)$$

4 Farbe

4.1 Color

Bearbeitungszeit: 30 min

Die Klasse `Color` repräsentiert eine Farbe im RGB-Farbraum. Die Farben sollen einen Wert zwischen 0 und 1 haben.

Im Konstruktor werden die RGB-Kanäle einzeln als `double` Werte übergeben.

Es wird überprüft, ob die übergebenen Werte innerhalb des zulässigen Wertebereichs (0 - 1) liegt. Ist dies der Fall, werden die Kanäle als öffentliche Klassenattribute gespeichert. Sollten die Werte über- oder unterhalb des Wertebereichs liegen, werden die Werte auf 0 bzw. 1 gesetzt.

Die Klasse verfügt über Methoden zur Addition, Subtraktion und Multiplikation mit einer anderen Farbe, die jeweils ein neues Objekt der Klasse `Color` zurück geben.

Außerdem gibt es eine Methode, um eine Farbe mit einem `double` Wert zu multiplizieren.

5 Geometrie

5.1 Geometry (abstrakte Klasse)

ca. 10 min

Die Klasse `Geometry` ist abstrakt und repräsentiert eine Geometrie im Raum. Im Konstruktor wird ihr eine Farbe zugewiesen. Außerdem deklariert sie eine Methode `Hit` `hit(final Ray r)`, um einen Schnittpunkt zwischen ihr und einem Strahl `r` berechnen zu können. Die Implementierungen dieser Methode ist in jeder Geometrie individuell zu implementieren.

5.2 Hit

Bearbeitungszeit: 5 min

Die Klasse Hit repräsentiert einen Schnitt zwischen einem Strahl und einer Geometrie.

5.3 Plane

Bearbeitungszeit: 30 min

Die Klasse Plane repräsentiert eine Ebene im Raum. Dabei wird sie aus einem Punkt und einer Normalen gebildet. Sie besitzt zudem eine Farbe, die über Geometry festgelegt wird. Die Methode Hit `hit(final Ray r)`, gibt den Schnittpunkt zwischen der Ebene und einem Strahl `r` zurück. Existiert kein Schnittpunkt, so gibt die Methode null zurück.

Schnittpunktberechnung:

$$t = \frac{(\vec{a} - \vec{o}) \cdot \vec{n}}{\vec{d} \cdot \vec{n}} \quad (6)$$

5.4 Sphere

Bearbeitungszeit: 30 min

Die Klasse Sphere repräsentiert eine Kugel im Raum. Dabei wird sie aus einem Punkt und einem gegebenen Radius gebildet. Sie besitzt zudem eine Farbe, die über Geometry festgelegt wird. Die Methode Hit `hit(final Ray r)`, gibt den kleinsten positiven Schnittpunkt zwischen der Kugel und einem Strahl `r` zurück. Existiert kein Schnittpunkt, so gibt die Methode null zurück.

Schnittpunktberechnung: (Formeln siehe Folien aus dem SS14 zu ebene/kugel S.10)

$$a = \vec{d} \cdot \vec{d} \quad (7)$$

$$b = \vec{d} \cdot (2 \cdot (\vec{o} - \vec{c})) \quad (8)$$

$$c = (\vec{o} - \vec{c}) \cdot (\vec{o} - \vec{c}) \cdot r^2 \quad (9)$$

$$t = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (10)$$

$$(11)$$

6 World

6.1 World

Bearbeitungszeit: 40 min

Die Klasse World repräsentiert eine Szene in der Geometrie und Strahl vorkommen.

Die Methode `Hit hit (final Ray r)` gibt den kleinsten positiven Schnittpunkt zwischen einer Geometrie und dem zu prüfenden Strahl aus. Gibt es keinen Schnittpunkt, so wird die Hintergrundfarbe der Szene durch ihr Attribut `backgroundColor` festgesetzt (in unserem Fall schwarz).

7 RayTracer

7.1 RayTracer

Bearbeitungszeit: 6 Stunden, aufgrund von Fehlersuche und -behebung des Tests

Im Konstruktor wird ein `ImageCanvas` erzeugt und vorbereitet.

7.2 TestRay

Erstellt 4 verschiedene Welten und lässt sie vom Raytracer erstellen.