

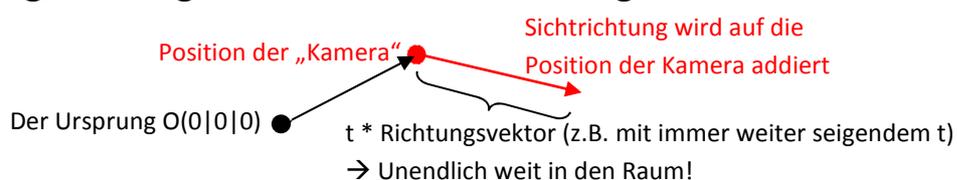
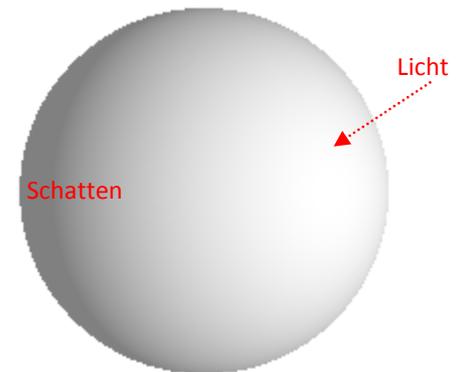
C# – Programm: „Raytracer (3D – Renderer)“

Hiermit verbrachten wir die letzte Einheit in C# des Informatikunterrichtes. Dieser Raytracer ist ein Programm, das nur mit wenigen Informationen über einen Raum, der geometrische Körper enthält, ein realistisches Bild berechnet. Dabei hält er sich, soweit wir es definiert haben, an die mathematisch und physikalisch (weitestgehend) korrekten Eigenschaften von „Photonen“, also Lichtstrahlen. Es können Kugeln, Zylinder und selbst kreierte 3D – Objekte im .obj – Format in die Szene eingebracht werden. Außerdem gibt es vier verschiedene Lichttypen, so viele Materialien wie Sie wollen, eine ausgerichtete Kamera und einen Attribute – Manager, der die Verwaltung übersichtlich und einfach gestaltet.

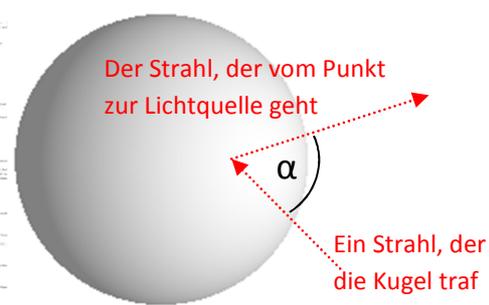
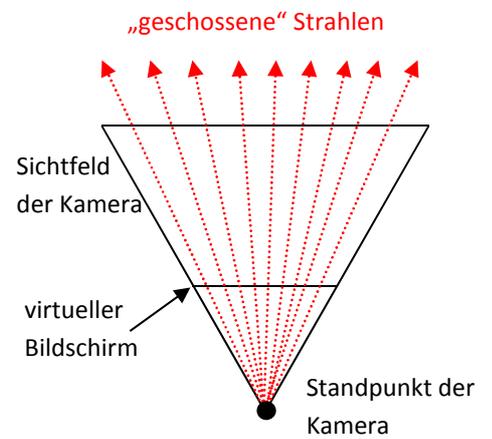
Dokumentation dieses Programmes:

Zu allererst möchte ich etwas Allgemeines zum Verständnis des Raytracings erklären: Es gibt einen Raum, der z.B. eine Kugel und eine punktförmige Lichtquelle enthält. Natürlich wird die Kugel von der Lichtquelle nicht überall gleich stark angestrahlt und deshalb hat sie dunklere und hellere Seiten.

Woher das Programm nun weiß, wo genau es heller und wo genau es dunkler ist, ist natürlich die Frage. Die Antwort ist das „Raytracing“, also „Strahlenverfolgung“. Eine genaue, jedoch rechenaufwändige Vorgehensweise. Dabei beschreiben wir zunächst alles, was wir haben, in mathematischer Form: Die sichtbare Hülle der Kugel besteht dann aus allen Punkten, die genau den Abstand des Radius der Kugel vom Mittelpunkt der Kugel haben. Der Strahl, den wir in den Raum schicken, lässt sich als Gerade beschreiben: Der Startpunkt (die Position unserer Kamera) + $t \cdot$ der Richtungsvektor von der Kamera weg, also eigentlich die „Sichtrichtung“ in die wir schauen.



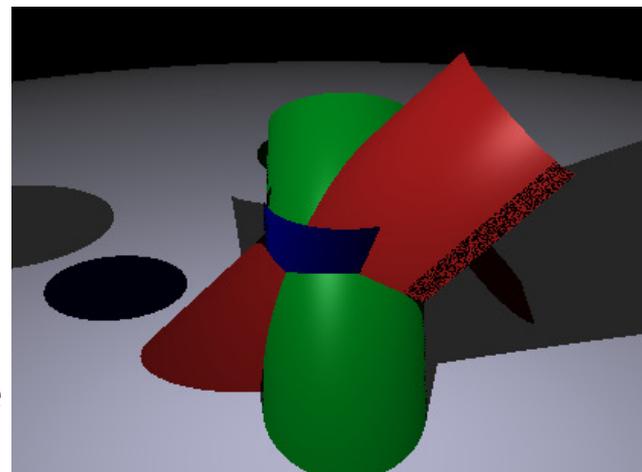
Nun schicken wir für jedes Pixel in unserem Bild, das wir nachher haben wollen, einen Strahl in den Raum. Dafür müssen wir wissen, wie unsere Kamera aufgebaut ist: Es gibt im Blickfeld der Kamera einen kleinen „virtuellen Bildschirm“, durch den von der Kamera aus in relativen Abständen zu der Anzahl der Pixel im Bild Strahlen geschossen werden. Nun müssen wir jedes Objekt im Raum prüfen, ob der Strahl es schneidet. Mit verschiedenen mathematischen Werkzeugen kann der Schnittpunkt mit der Kugel errechnet werden. Nun wissen wir, bei welchem t des Strahles die Kugel sichtbar wird. Um jetzt besondere optische Effekte zu



erhalten, wird zuerst zu jeder in der Szene vorhandenen Lichtquelle ein weiterer Strahl gesendet, der nun jedoch vom Durchstoßpunkt in der Kugelhülle aus geht. Nachdem der Strahl feststeht, der von der Kugel aus geht, wird der Winkel α zwischen

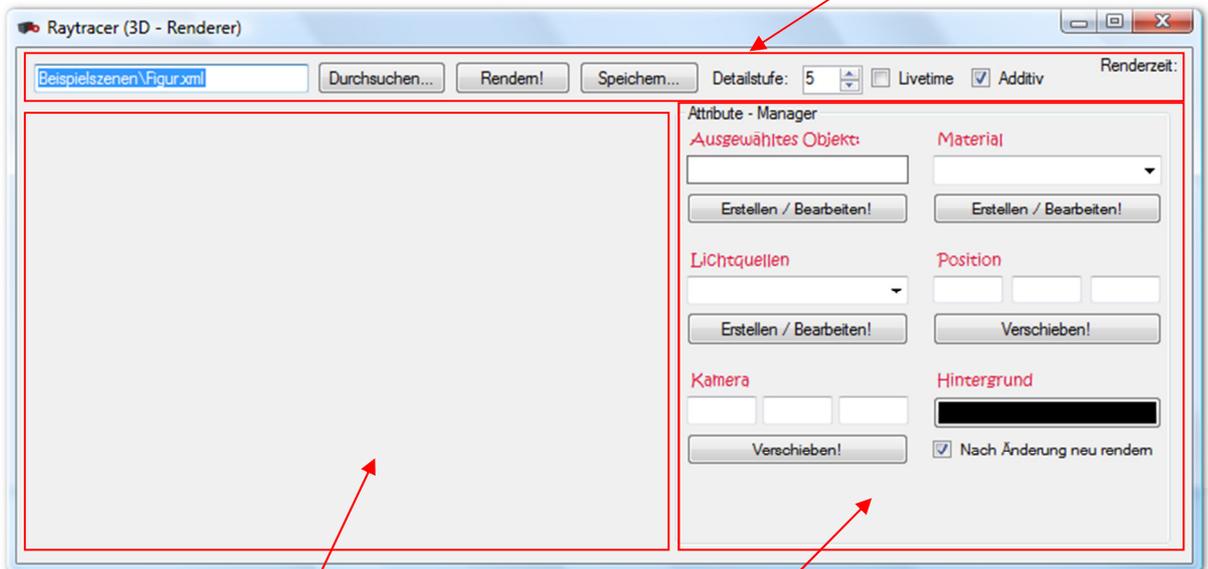
den beiden Strahlen berechnet. Je größer α wird, desto schräger kommt der Strahl von der Lichtquelle zu diesem Punkt. Und je schräger ein Lichtstrahl auf einen Punkt trifft, desto weniger Energie, die z.B. in Lichtenergie umgewandelt werden kann, kommt an und desto weniger hell ist dieser Punkt. Alle diese Begebenheiten werden in die Rechnung miteinbezogen und zusammenaddiert, bis man schließlich den Farbwert für das eine Pixel hat, wegen dem man den Strahl durch den virtuellen Bildschirm schickte. Natürlich gibt es noch viele weitere Faktoren, die auf die Rechnung einwirken oder einen völlig anderen Ansatz benötigen (z.B. reflektierendes oder transparentes Material).

Doch nun werden Sie die Bedienung des Programmes kennenlernen, nachdem Sie erfahren haben, wie Raytracing ungefähr funktioniert. Die Programmierung, bei der man sich mit wirklich allen Faktoren auseinandersetzen muss, ist natürlich viel komplizierter, sodass in manchen Entwicklungsphasen des Programmes gerne einmal „moderne Kunst“ entstand...

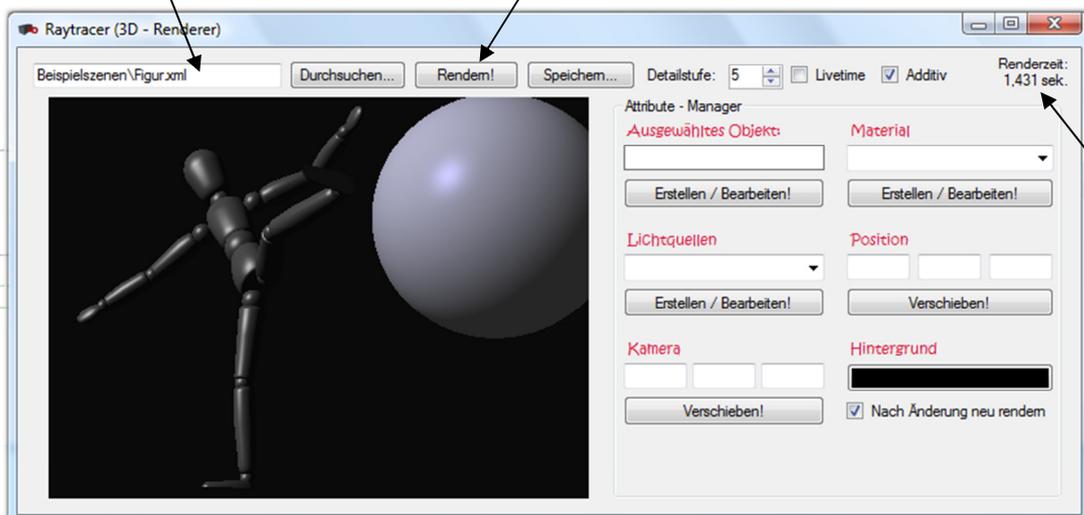


1. Der Start in eine virtuelle Welt (Rendern)

Wenn Sie mein Programm starten, sehen Sie ein Interface, das in drei Teile geteilt ist. Der erste Teil ist der Dateiverwaltungsteil und auch der Hauptbedienungsteil.



Es folgen der Ausgabeteil und der Attribute – Manager. Der zunächst vielleicht nicht als solcher erkennbare Ausgabeteil ist am Anfang leer: Es wurde ja auch noch kein Bild berechnet. Der Attribute – Manager wird im Verlauf der Dokumentation noch genauer erklärt, da er sämtliche Steuerungen der Szene enthält und für die Bearbeitung von dieser wichtig ist. Zuerst wollen wir jedoch noch nichts bearbeiten, sondern einfach einmal eine Szene als Bild rendern (also berechnen) lassen! Dafür bietet sich die Beispielszene „Figur“ an, die ich zu Testzwecken erstellt habe. Sie steht schon standardmäßig in der Box für den Projektpfad. Drücken Sie „Rendern!“, um die Szene zu rendern.



Die für die Berechnung benötigte Zeit

Nun sehen Sie schon nach kurzer Berechnungszeit das fertige Bild. Sie haben außerdem die Möglichkeit, die Option „Livetime“ zu aktivieren: Hierbei sehen Sie sogar, wie lange der Renderer für welche Stelle des Bildes ungefähr braucht, da das Bild sich dann aufzeichnet. Obwohl die Funktion sehr anschaulich ist, hat sie den Nachteil, dass Sie etwas langsamer als das normale Rendern ist, denn fertige Pixelmengen müssen noch während der Berechnung ausgegeben werden. Wenn Sie nicht wollen, dass bei jedem Mal livetime – Rendern der Ausgabeteil geleert wird, können Sie die Option „Additiv“ aktivieren, mit welcher die neuesten Renderergebnisse einfach **über** das alte Bild gezeichnet werden. Außerdem können Sie den Detailgrad numerisch einstellen. Dieser bedeutet Folgendes:

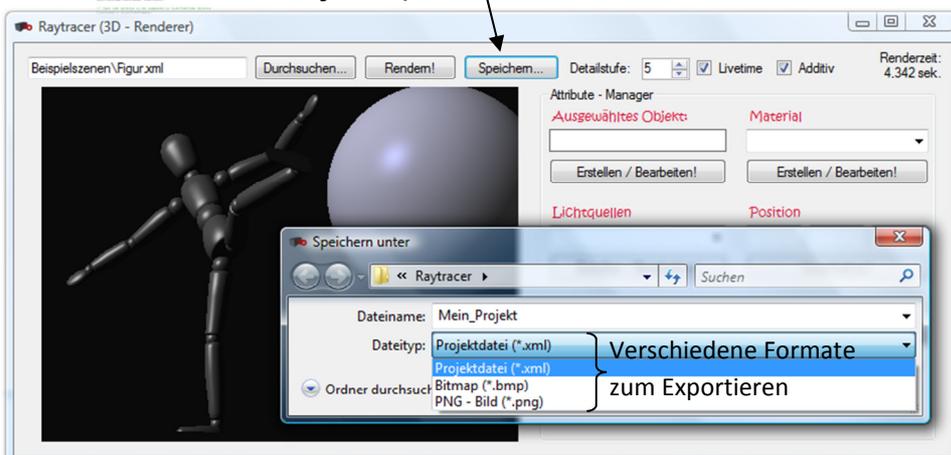
1. Grad	Es werden nur Umrisse erkannt und entsprechend eingefärbt.
2. Grad	Diffuse Umgebungsbeleuchtung lässt die Objekte plastisch und räumlich wirken.
3. Grad	„Specular Highlights“, also Glanzpunkte bewirken, dass Objekte echter erscheinen.
4. Grad – 8. Grad	Pro Grad wird eine Spiegelung mehr pro Pixel verfolgt. (Wenn mehrere spiegelnde Objekte in der Szene vorhanden sind)

Es lässt sich sagen: Je detailreicher, desto länger benötigt das Programm, um die Szene zu rendern.

2. Sichern der Ergebnisse (Speichern)

Sie sollten Ihr Projekt regelmäßig speichern. Dies geht über die „Speichern...“ – Schaltfläche. Sie können auswählen, ob Sie das Projekt (also die XML – Datei mit den einzelnen Objekten, den

Lichtern und den Materialien) oder ob Sie das gerenderte Bild als Bitmap oder PNG – Bilddatei speichern wollen.



3. Die Kreativität walten lassen (Erstellen)

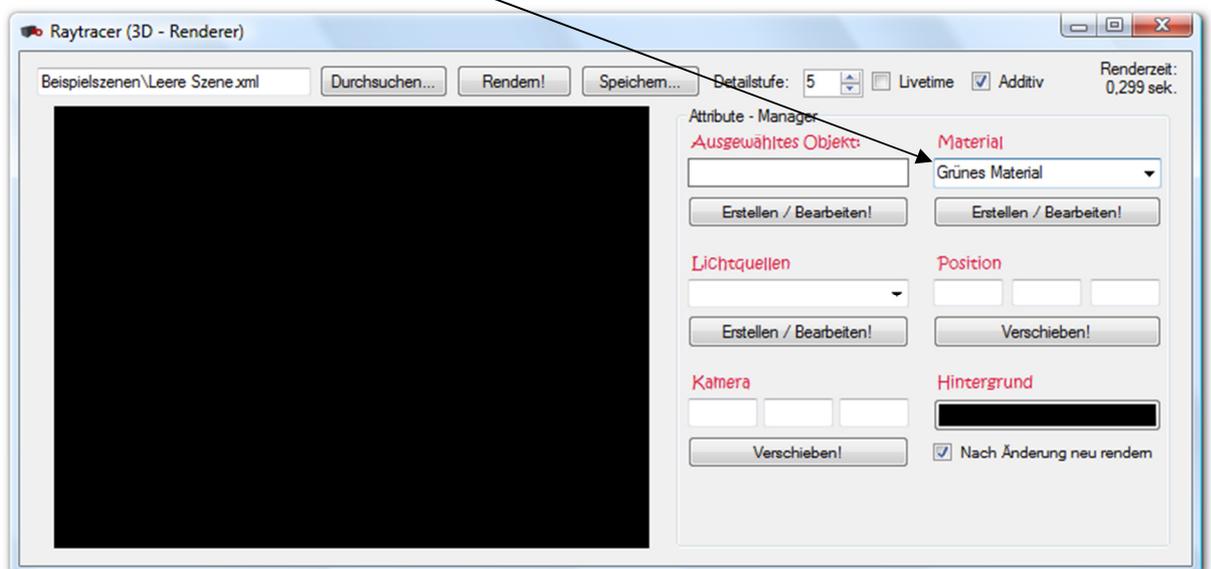
Nachdem Sie jetzt auch wissen, wie Sie schnell und einfach Szenen rendern können, geht es an die Szene selbst!

Grundsätzlich können Sie alles entweder erstellen oder es, wenn es schon in der Szene existiert, bearbeiten.

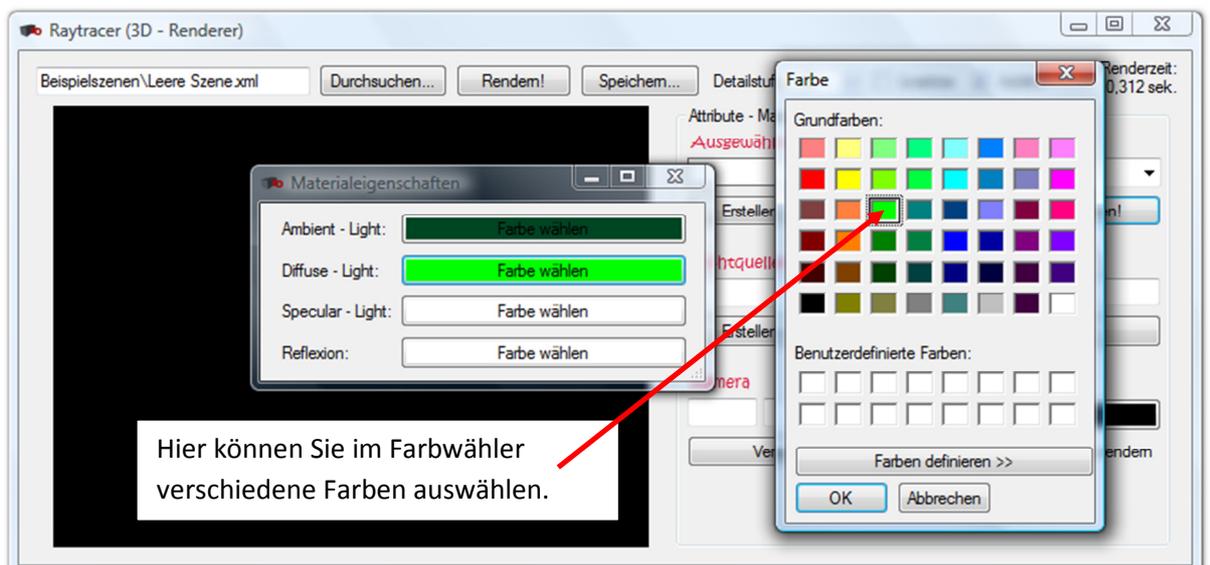
a) Materialien

Zuerst befassen wir uns mit dem Erstellen von Materialien, denn diese müssen schon vorliegen, wenn wir ein Objekt erstellen. Laden Sie also meine extra für Kreative angehängte Projektdatei „Leere Szene.xml“, die nichts außer die Kamera und ein paar Grundeinstellungen enthält.

Geben Sie dann einen Namen für das neue Material ein.



Wenn Sie nun auf „Erstellen / Bearbeiten!“ klicken, zeigt sich ein Dialog, der genauere Eingaben zum Material erlaubt.



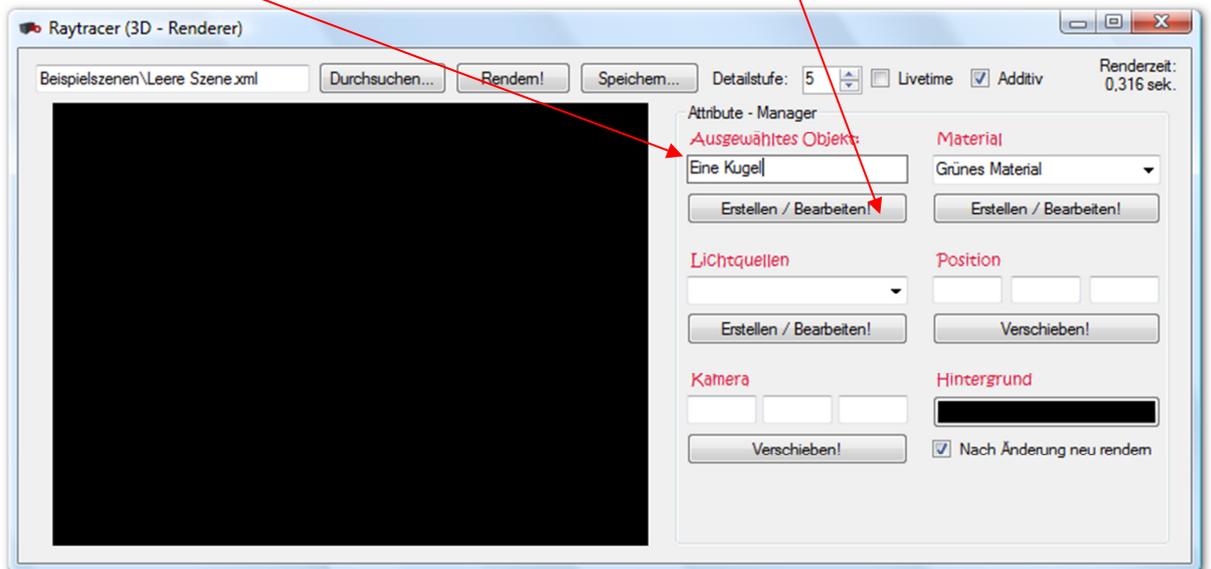
Die Farbanteile haben jeweils folgende Bedeutung:

Farbanteil	Bedeutung
Ambient – Light	Die allgemeine „Farbstimmung“ auf den Objekten der Szene.
Diffuse – Light	Der Hauptfarbton für dieses Objekt, also Materialfarbe des Objekts.
Specular – Light	Die Farbe für die kleinen Glanzpunkte auf Objekten. (z.B. glatte Oberfläche)
Reflexion	Wie stark dieses Material seine Umgebung reflektiert.

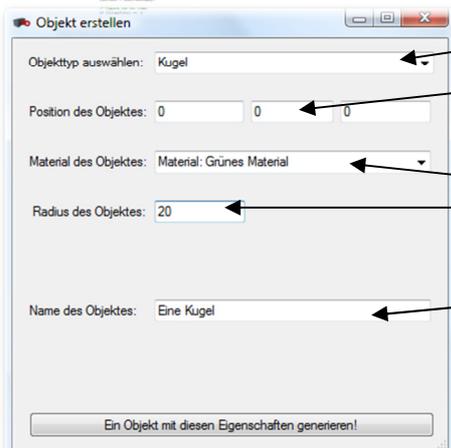
Wenn Sie zufrieden mit Ihrem Material sind, können Sie das Fenster schließen. Um das Material auch anwenden zu können, empfiehlt es sich, auch ein Objekt zu erstellen:

b) Objekte

Um ein neues Objekt zu erstellen, geben Sie den gewünschten Objektnamen ein und klicken auf „Bearbeiten / Erstellen!“

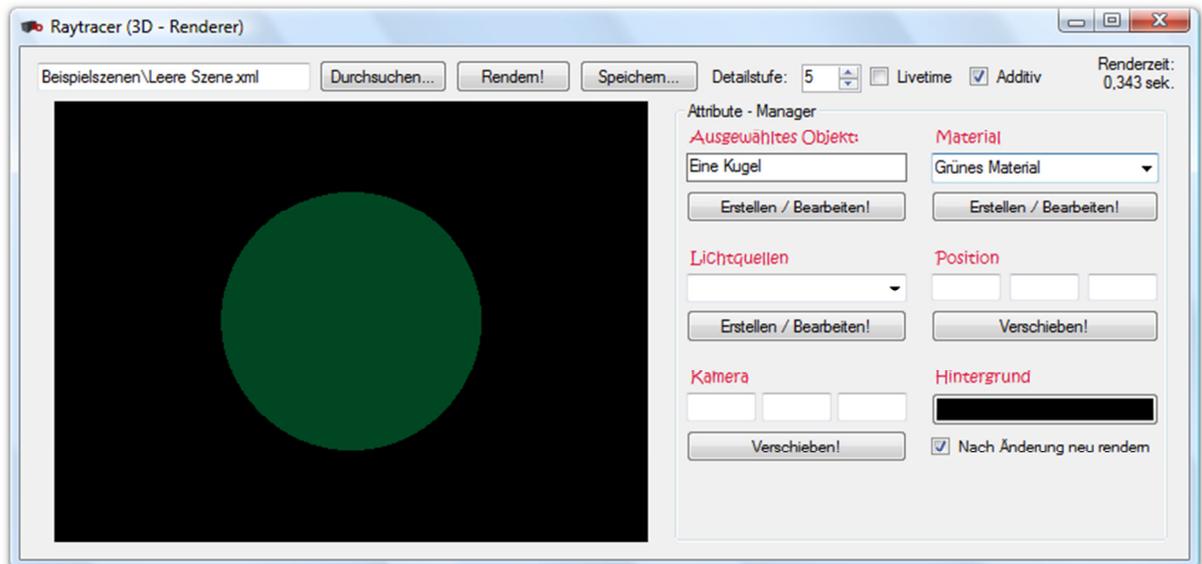


Auch hier öffnet sich ein Dialog, der genaueres über das Objekt wissen möchte. Füllen Sie diesen vollständig aus.



Der Objekttyp ist zuerst gefragt. Dann wird die Position benötigt, bei der sich (hier z.B.) Kugel befindet. Jetzt können Sie auch das vorher erstellte Material aus der Auflistung auswählen. Für die Kugel wird noch der Radius benötigt. Der Name ist bei jedem Objekt anzugeben.

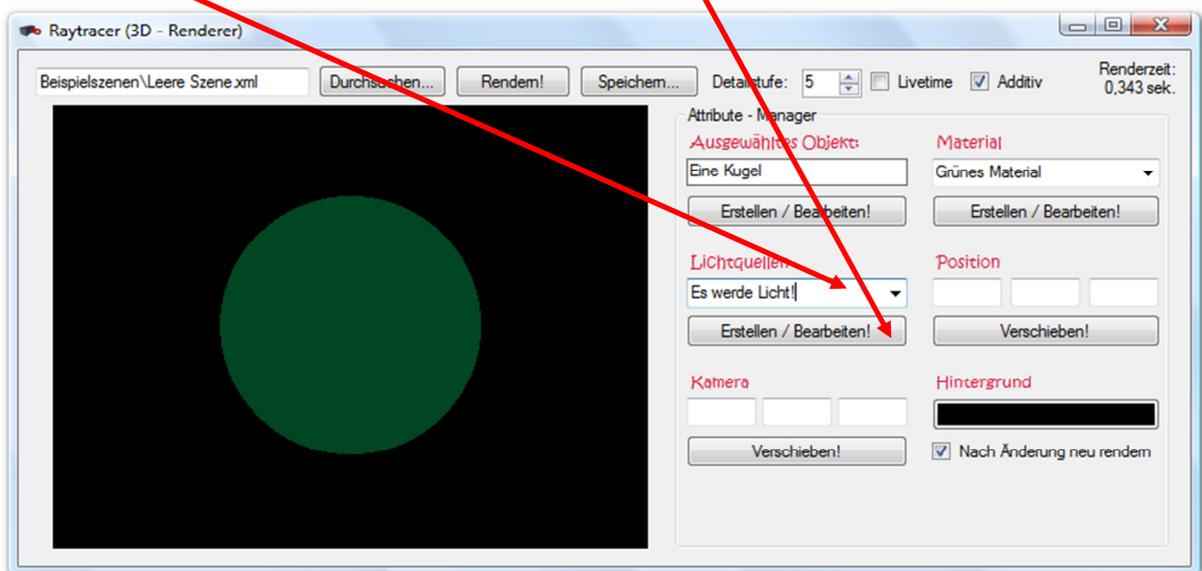
Wenn Sie alles eingegeben haben, klicken Sie auf „Ein Objekt mit diesen Eigenschaften generieren!“ und das Objekt wird sofort mit in die Szene miteinbezogen.



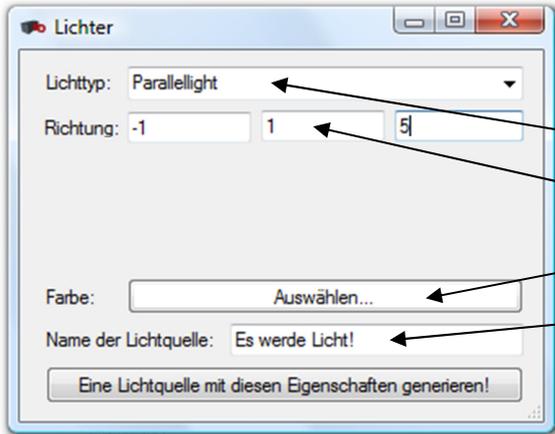
Etwas seltsam ist, dass die Kugel einfach eine Fläche ist. Das liegt daran (wie schon in der Einführung angesprochen), dass jedes Objekt eine Lichtquelle benötigt, um dreidimensional zu wirken und eine plastische Oberfläche zu besitzen.

c) Lichtquellen

Eine neue Lichtquelle ist einfach zu erstellen: Man muss nur den Namen eingeben und auf „Erstellen / Bearbeiten!“ klicken.



Wieder erscheint ein Dialog, der nun genaueres über unsere neue Lichtquelle wissen möchte. Es gibt mehrere Typen von Lichtquellen, die von diesem Programm unterstützt werden.

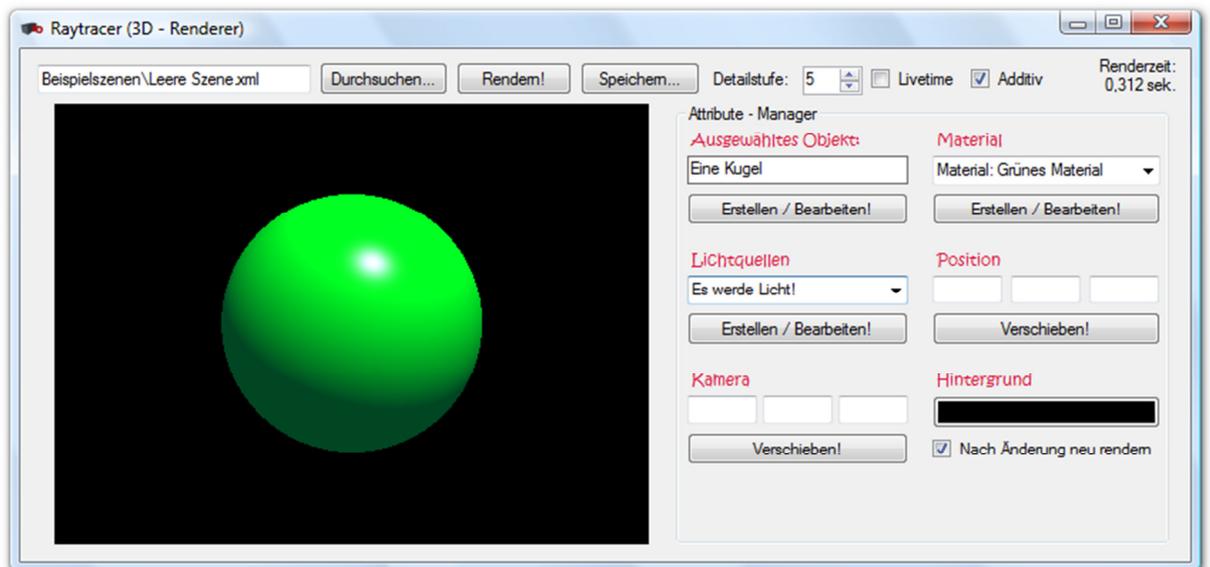


Auch hier ist wieder zuerst ein Typ gefragt. Dieses ParallelLight – Licht kommt aus der eingegebenen Richtung mit der weißen Farbe (volle Kraft) und trägt den Namen „Es werde Licht!“, der natürlich nur einmal vorkommen darf. Wenn alles konfiguriert ist, können Sie auf „Eine Lichtquelle mit diesen Eigenschaften generieren!“ klicken.

Vorher noch ein kurzer Überblick über die Lichttypen:

Lichttyp	Besonderheiten
FalloffLight	Die Intensität dieses Lichtes nimmt mit zunehmender Entfernung der Lichtquelle ab.
OmniLight	Dieses normale Licht strahlt einfach in alle Richtungen gleichstarkes Licht ab.
ParallelLight	Hier hat das Licht keine Position, sondern nur eine Richtung, aus der es zu jedem Punkt hin eintrifft. (Vergleiche z.B. Sonnenlicht)
SpotLight	Dieses Licht hat einen Zielpunkt, zu welchem es gerichtet ist und außerdem eine Winkelbegrenzung, die einen kegelförmigen Schein erzeugt. (Vergleiche z.B. Scheinwerfer)

Wenn Sie also bestätigt haben, wird auch das Licht ab jetzt mit in die Szene miteinbezogen:



Und schon erscheint die Kugel sehr viel plastischer!

4. Die Szene ausfeilen und anpassen (Bearbeiten)

Natürlich lassen sich alle Attribute auch wieder verändern. Die Dokumentation bezieht sich dabei jedoch auch auf **3.**, denn dort ist der Umgang mit den Dialogen schon geklärt worden.

a) Materialien

Wählen Sie einfach das Material aus, klicken Sie auf „Erstellen / Bearbeiten!“ und konfigurieren Sie das Material neu. (Umgang mit Dialog: s. **3.**)

b) Objekte

Hier kommt eine besondere Funktion zum Tragen: Klicken Sie einfach auf ein Objekt, das Sie im gerenderten Bild sehen und schon wird das Objekt im Attribute – Manager angezeigt. Durch einen Klick auf „Erstellen / Bearbeiten!“ können Sie das Objekt bearbeiten. (Umgang mit Dialog: s. **3.**)

c) Lichtquellen

Wählen Sie auch hier einfach eine Lichtquelle aus und klicken Sie auf „Erstellen / Bearbeiten!“, um die Lichtquelle neu einzustellen. Übrigens kann sogar der Typ der Lichtquelle gewechselt werden. (Umgang mit Dialog: s. **3.**)

Ich hoffe, dass Sie noch viel Spaß mit dem Raytracer haben werden und Ihre Experimentierfreude Sie vielleicht zu dem ein oder anderen erstaunlichen Ergebnis bringen wird!

Ich empfehle auch die vielen anderen Szenen im beigelegten „Beispielszenen“ – Ordner zum Ausprobieren.



Das Gerüst für den Raytracer (diverse Klassen, Grundideen und „Infrastrukturen“) wurde von meinem Informatiklehrer Rainer Helfrich programmiert!

5. Versionsgeschichte des Programmes

➔ Die aktuellste Version 1.0.0.0 erschien am 28.04.2013

a) Alles wurde bearbeitbar gemacht

b) Das Interface wurde neu aufgelegt und optimiert

6. Rechtliche Hinweise:

Das komplette Programm mit allen Ideen, Grafiken, Programmabläufen und sonstigem geistigen Material ist geschützt und darf nicht kopiert, verändert oder veröffentlicht werden (außer von Johannes Schirm). Johannes Schirm haftet für keinerlei Schäden, die durch die Überschreibung und Löschung von Variablen oder falsche Verarbeitung des Programmes entstehen. Es ist ausdrücklich **nicht** erlaubt, diese Anleitung oder eines meiner C# – Programme im Internet ohne die Genehmigung von Johannes Schirm zu veröffentlichen. Bei Fragen, Kritik, Lob und Anregungen bitte ich um eine Nachricht („Kontakt“) über meine Website.

www.johannes-schirm.de

Vielen Dank für Ihr Interesse an meinen C# – Programmen!

(Sehen Sie auch die anderen interessanten Sachen auf meiner Seite an und hinterlassen Sie einen Gästebucheintrag!)