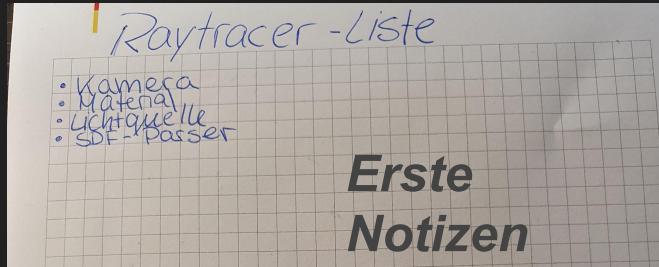


# Raytracer

Marie Trojan und Fionn E. Erickson

# Erste Schritte

- Überlegung der ersten Gedanken
  - > erstellen des SDF-Parsers
  - > erstellen wichtiger Klassen  
(Camera, Light, Material)



Commits on Aug 24, 2020

- added light  
Mielein committed 17 days ago

Commits on Aug 20, 2020

- scene struct  
Mielein committed 21 days ago
- Revert "edit camera"  
Mielein committed 21 days ago
- Revert "konstruktoren für die kamera"  
Mielein committed 21 days ago
- Revert "konstruktoren für die kamera edit"  
Mielein committed 21 days ago
- konstruktoren für die kamera edit  
Flonnn authored and Flonnn committed 21 days ago
- konstruktoren für die kamera  
Flonnn authored and Flonnn committed 21 days ago
- edit camera  
Flonnn authored and Flonnn committed 21 days ago

Commits on Aug 19, 2020

- material.hpp updated / light.hpp updated / basic camera.hpp  
Mielein committed 22 days ago
- material.hpp updated basic light.hpp  
Mielein committed 22 days ago
- basic material.hpp  
Mielein committed 22 days ago
- SDF-Parser implemented TO-DO's left  
Mielein committed 22 days ago

Commits on Aug 9, 2020

- initial commit  
Mielein committed on 9 Aug

# Erstellen von Boxen und Spheres

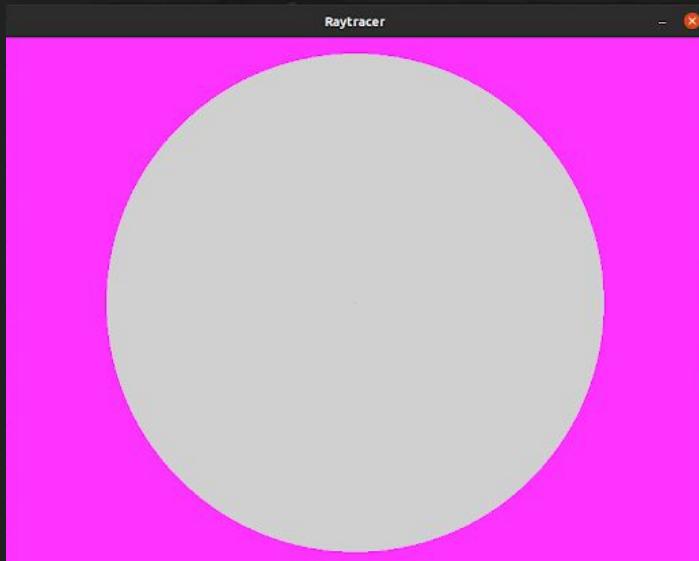
The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Aktivitäten < Visual Studio Code > Di 23:15 ● example.cpp - Untitled (Workspace) - Visual Studio Code
- File Explorer:** Shows the project structure under OPEN DATORS:
  - UNITLED WORKSPACE
  - glm-0.9.5.3
  - nanovg
  - framework
  - ambient.hpp
  - box.cpp
  - box.hpp
  - camera.hpp
  - checkerboard.hpp
  - CMakeLists.txt
  - color.hpp
  - example.hpp
  - filename
  - glew.c
  - hitpoint.hpp
  - light.hpp
  - material.hpp
  - pixel.cpp
  - pixel.hpp
  - plane.hpp
  - ppmwriter.cpp
  - ppmwriter.hpp
  - ray.hpp
  - renderer.cpp
  - renderer.hpp
  - scene.cpp
  - scene.hpp
  - shape.cpp
  - shape.hpp
  - sphere.cpp
  - sphere.hpp
  - tools.hpp
  - window.cpp
  - window.hpp
- Code Editor:** The main editor window displays the `example.cpp` file with the following code:

```
42     std::cout << " hier ist bullshit: " << sphere1.getMat()->ka << std::endl;
43
44     shape_vec.push_back(spheres1);
45     shape_vec.push_back(spheres2);
46     shape_vec.push_back(boxes1);
47     shape_vec.push_back(boxes2);
48
49     for(auto i : shape_vec){
50         std::cout << i << "\n" << std::endl;
51     }
52
53     Light light("stinkyLight",{-100.0f, 100.0 ,0.0f}, {0.5333f, 0.0f, 1.0f};
54     std::shared_ptr<Light> lights = std::make_shared<Light>(light);
55     light_vec.push_back(lights);
56
57     Camera cam();
58     Ambient amb("amb",{1.0,0.2,1.4});
59     Scene s(shape_vec,light_vec,mat_map,cam,amb);
60     Scene scene = sdfParser("../source/example.sdf");
61
62     for(auto a : s.light_vec){
63         std::cout << a->name_ << std::endl;
64     }
65
66     ppmwriter.cpp
67     ppmwriter.hpp
68
69     /* std::cout << "sdf read:<<std::endl;" */
70     /* renderer.render(scene); */
71     renderer.render(scene);
72
73
74     0x55f22459220
75     stinkyLight
76
77     Hallo Marie, ich rendere jetzt was, okay?
```
- Terminal:** Shows the command `0x55f22459220`.
- Bottom Status Bar:** master\* CMake: [Debug]: Bereit ➜ [GCC 7.5.0] Build [example] ➜ [example]

Zuerst keine  
großen Erfolge

# Erster Erfolg (irgendwie)



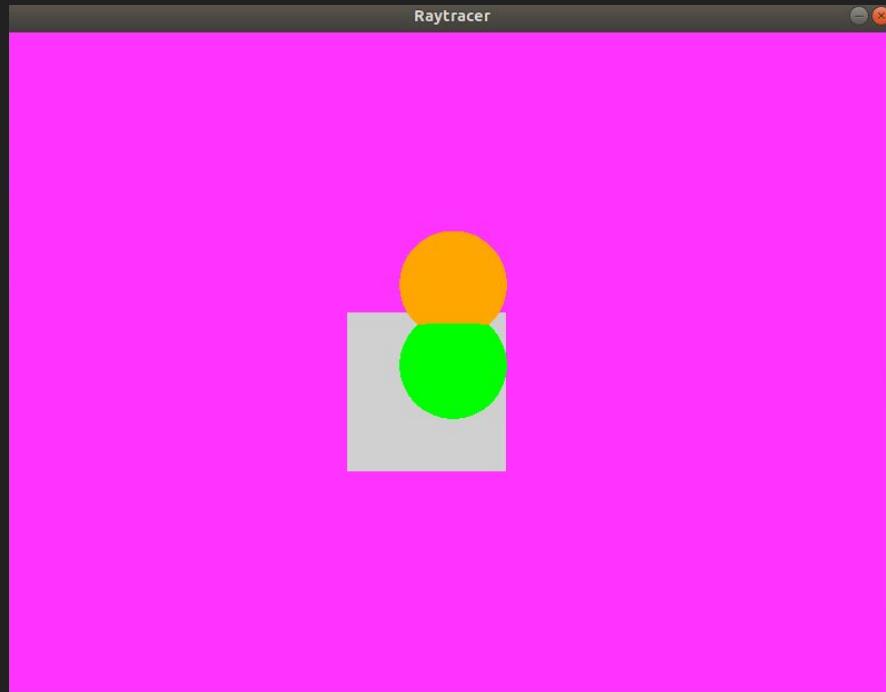
Intersect-Funktion funktioniert  
-> jedoch ist es die Box-Intersection

Wir wissen bis jetzt nicht warum, aber  
unsere Box wurde zu einem Kreis

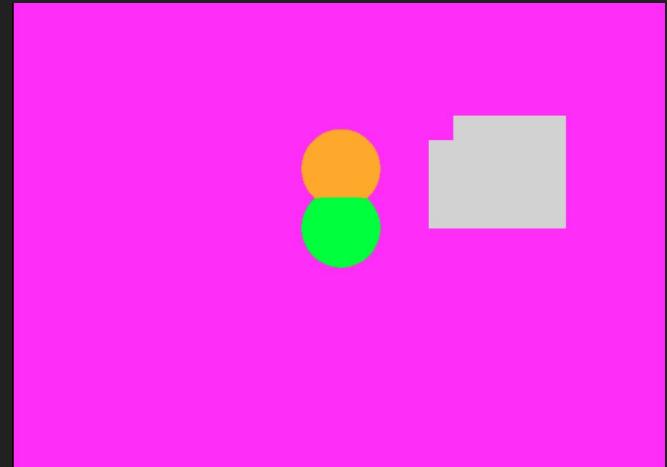


<- Moment als Fionn die "Box" sah

# Neue Box-Intersect sorgt für Hoffnung

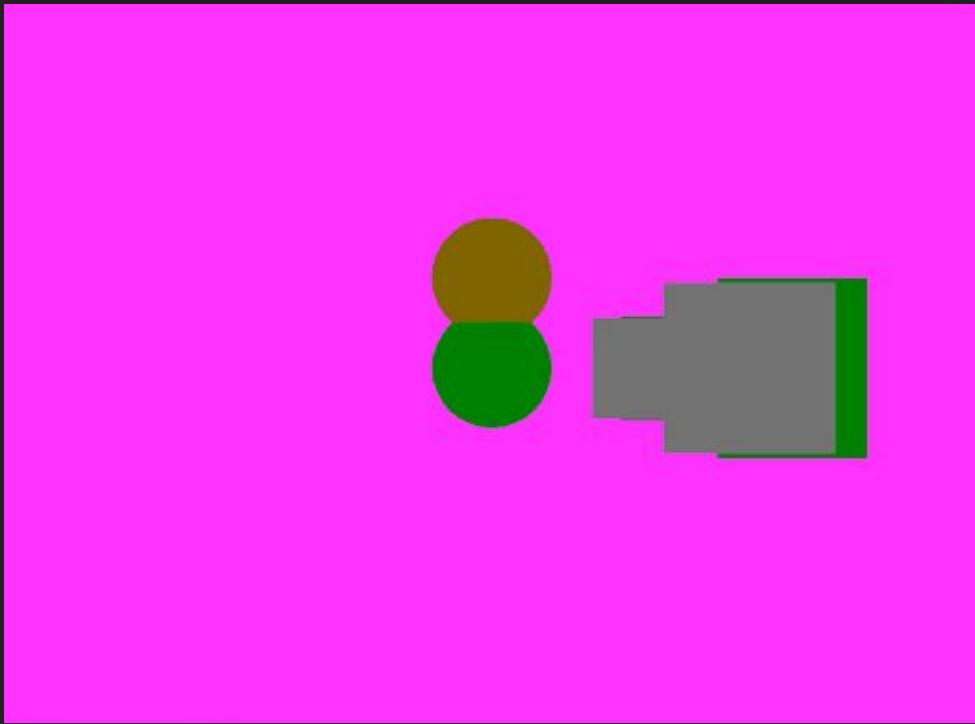


Box ist zu sehen, dank  
neuer Intersect.



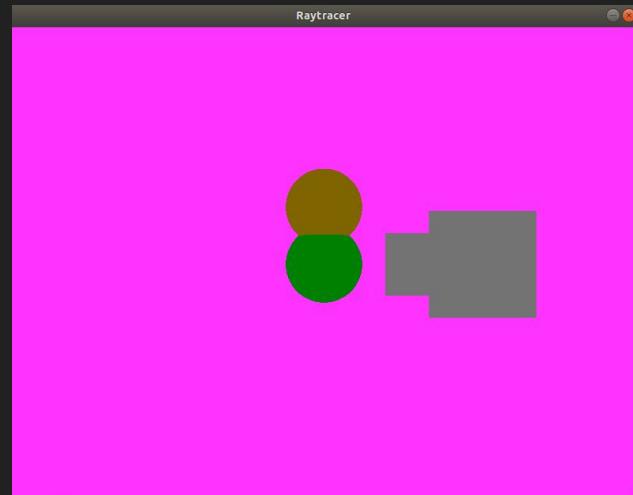
Box fehlen jedoch vier  
Seiten

# Tonemapping wird sichtbar

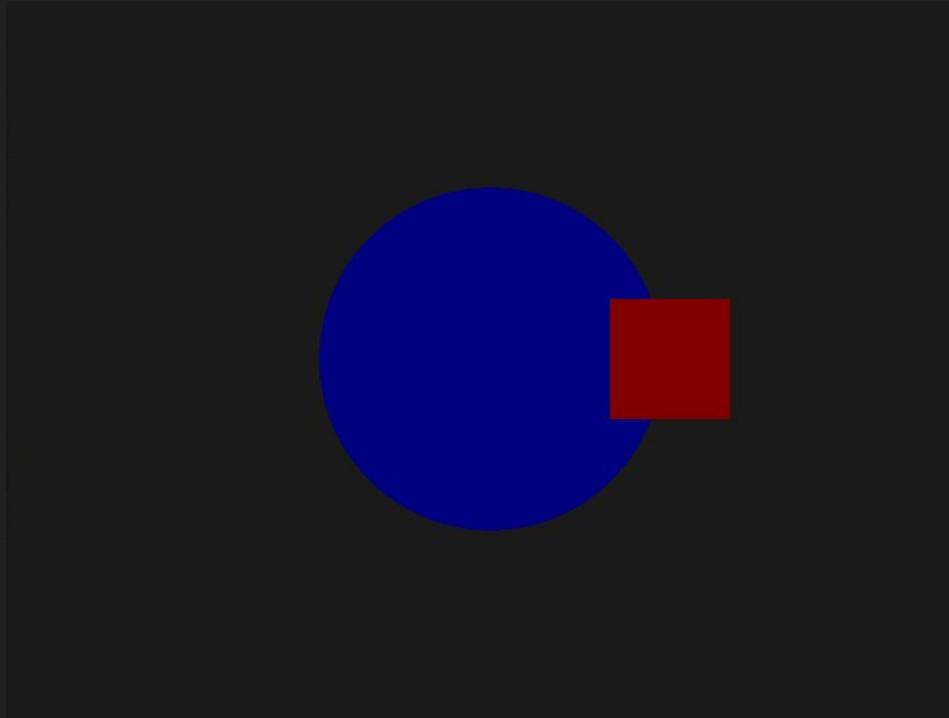


Durch verschieben der Box,  
wurde bestehendes Problem  
noch stärker sichtbar.

Zweite Box zeigte selbes  
Problem

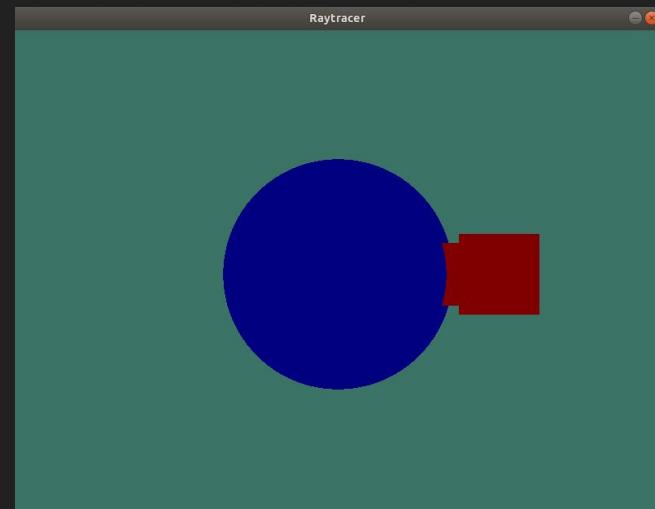


# Box debugging



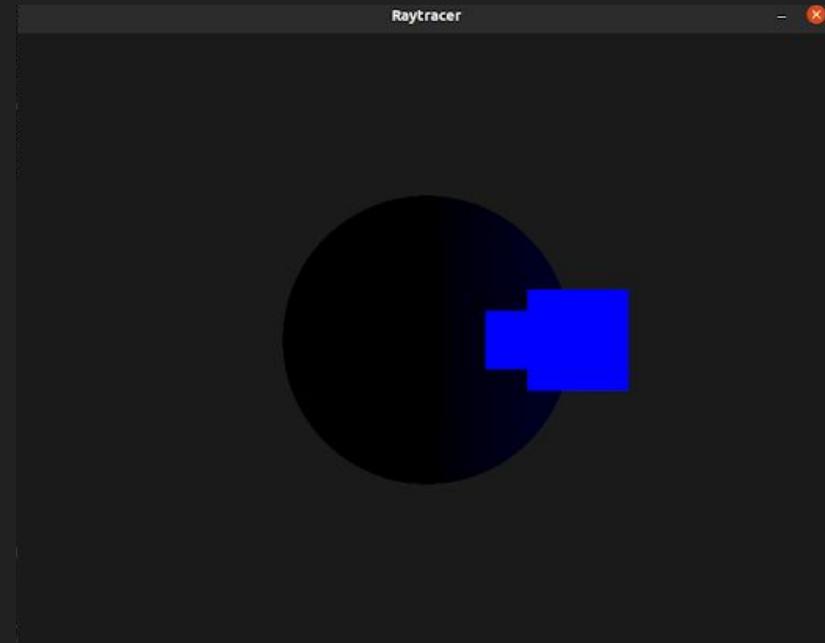
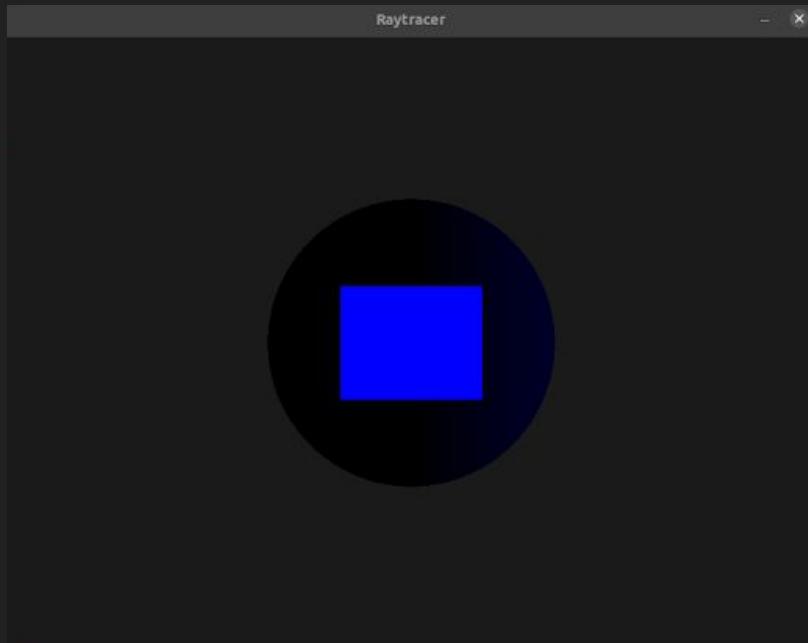
Box und Sphere eingelesen mit sdfParser (example.sdf)

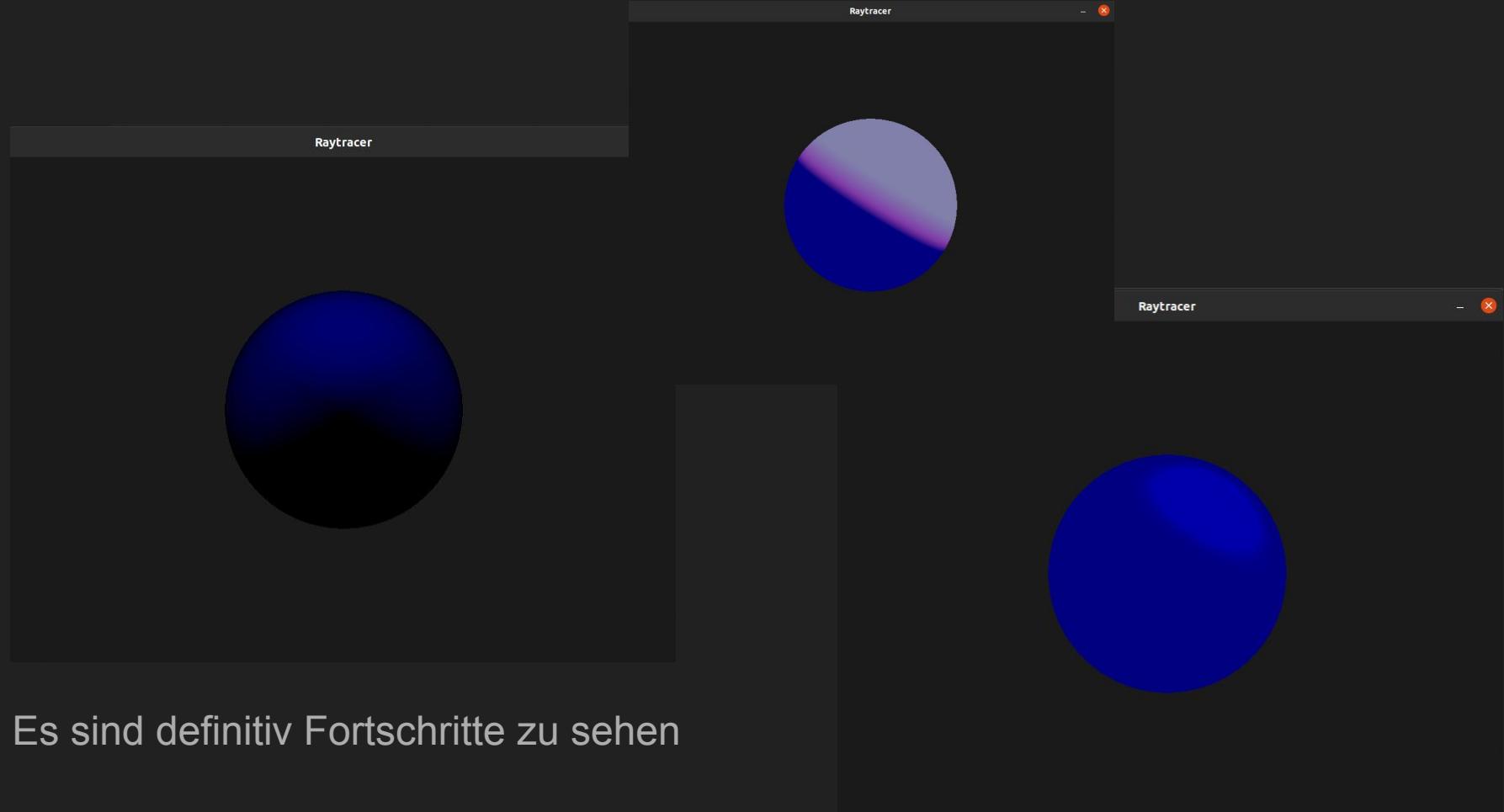
Langes debuggen mit einer Menge std::cout, sollen zur Problemlösung führen.



# Renderer

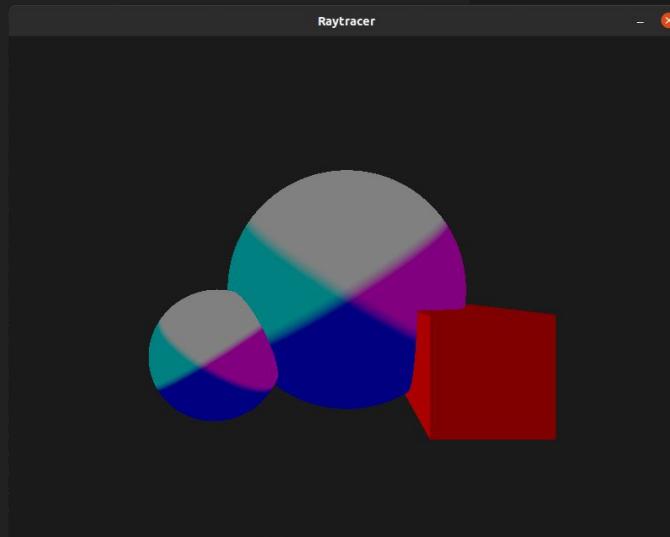
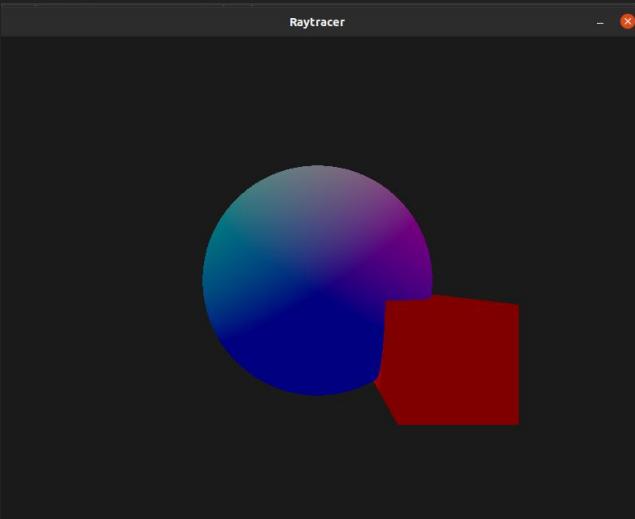
Nebenbei wurde in der render.cpp schon an dem shading gearbeitet



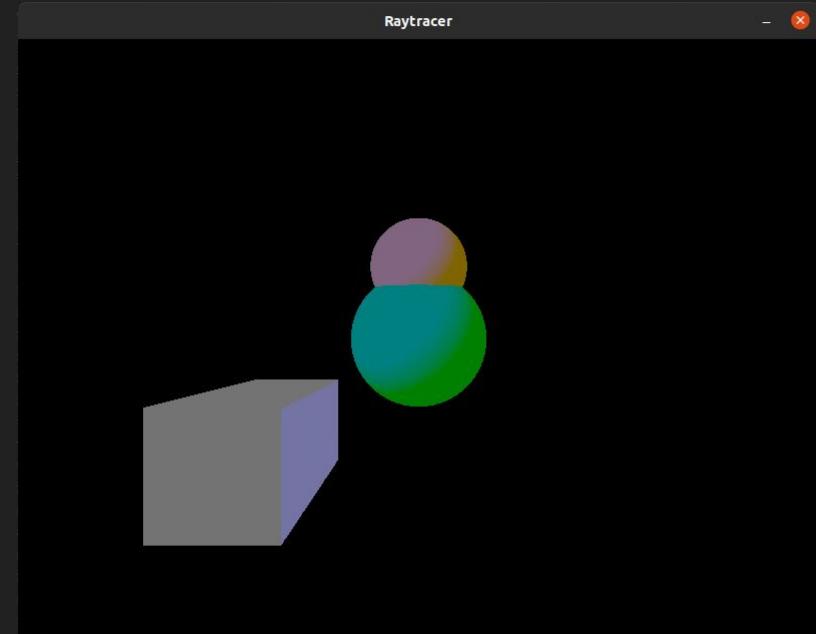


Es sind definitiv Fortschritte zu sehen

Es wurde viel rumexperimentiert, um  
Funktionsweise und bugs zu finden



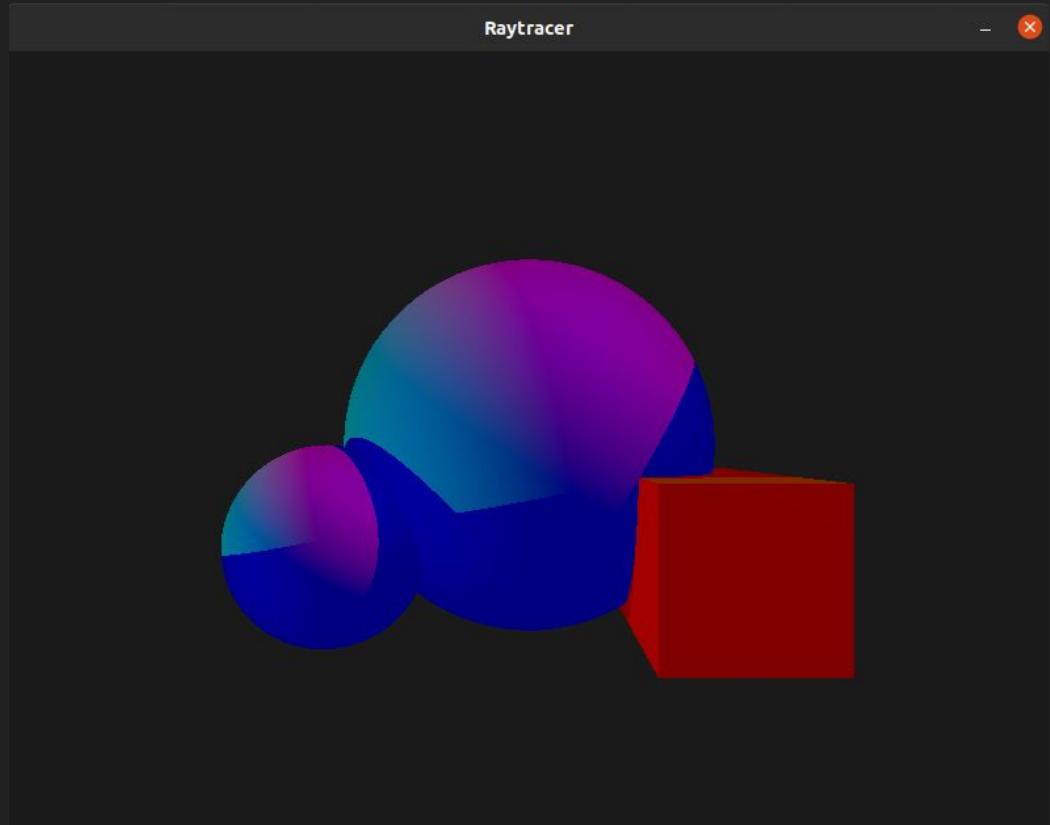
Beleuchtung in selbst erstellten Beispiel  
sah noch fragwürdig aus



# shadows work!



Not very happy about wrong shadowing tho



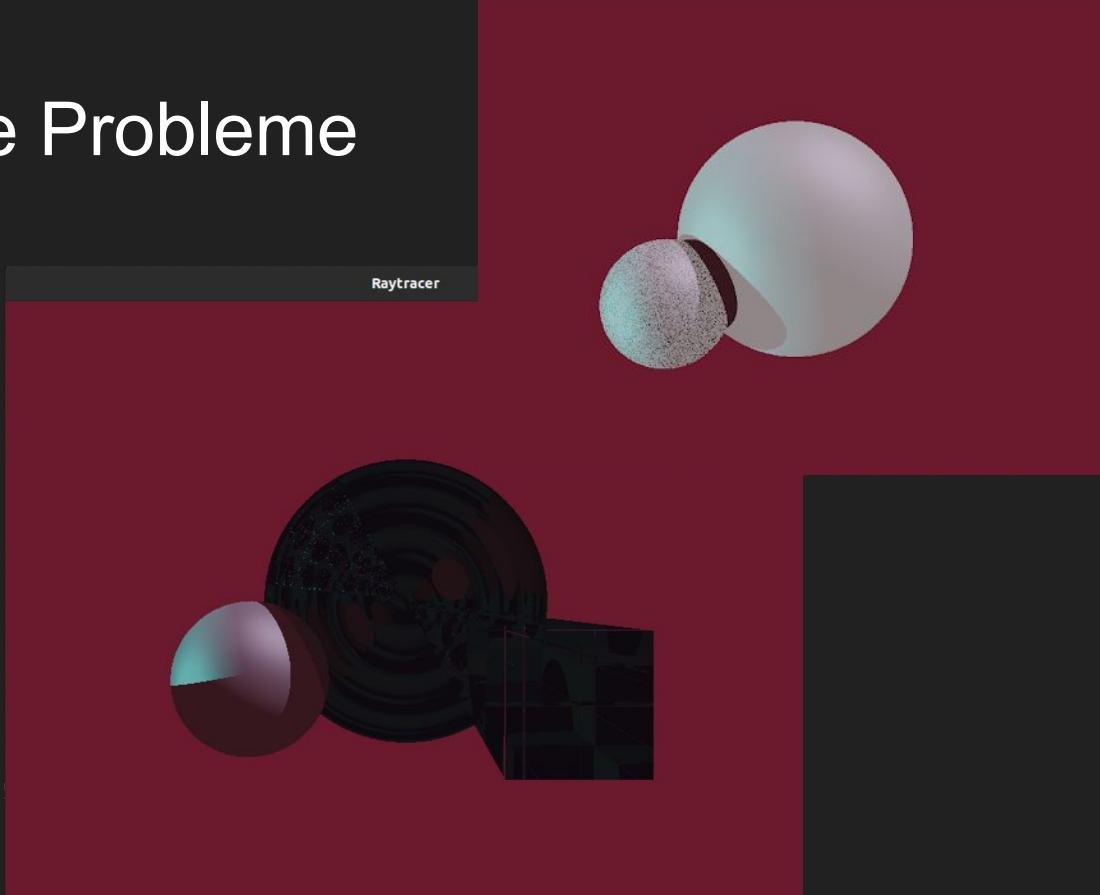
SHADOWS WORK!

unsere example.cpp sieht nun auch sehr  
cool aus :)

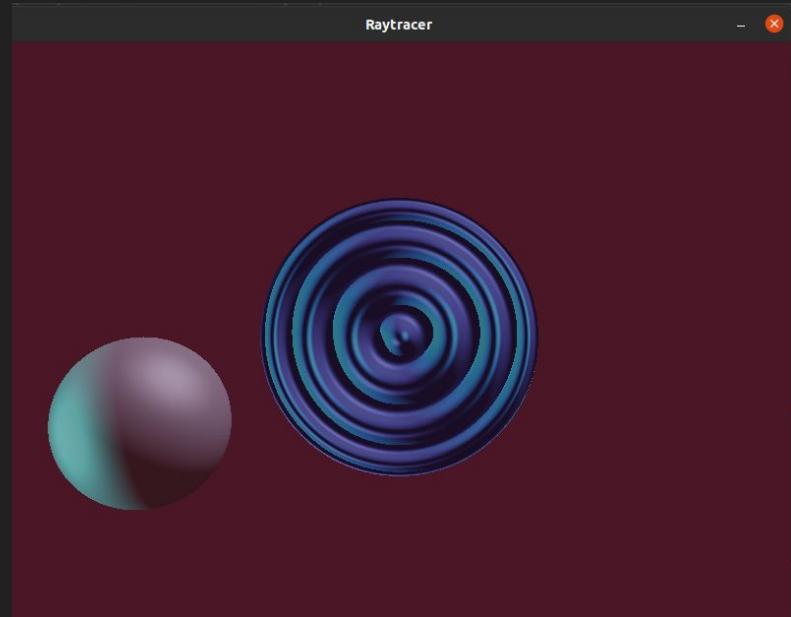


# Reflection bringt neue Probleme

Zeitgleich schafft auch  
die Transformation  
viele Sorgen



Spiegelung der Sphere  
(Normale zeigte in entgegengesetzte  
Richtung)

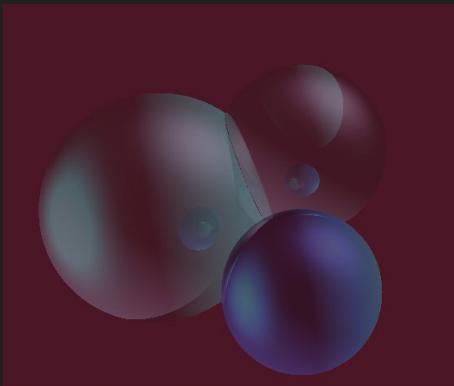


# Aufgabenteilung

- hauptsächlich zusammen gearbeitet
    - > jeder an anderer Aufgabe gearbeitet und bei Problemen Aufgaben getauscht und für Abwechslung gesorgt
  - Kommunikation fast täglich über Discord
- 
- grobe Aufteilung
    - Fionn: Renderer (Beleuchtungsmodell)
    - Marie: SDF-Parser / Körper

# Die Finalen Schritte

- wir haben die Reflektion hinbekommen
- und die Transformationen fertig implementiert



- Zu guter Letzt wollen wir noch die Kamera-Erweiterung in den sdf-Parser einbauen und eine Animation erstellen

# Die Rotation der Kamera und die Animation

- bei der Rotation der Kamera über den UP- Vector hatten wir noch ein paar Probleme, und uns nicht ganz klar geworden wieso es das bild verzerrt und nicht Angewinkelt hat.
- Bei der Animation haben wir uns dafür entschieden die Kamera durch einen Pfad gesäumt mit Sphären und Boxen welche unterschiedliche Materialien besitzen und beleuchtet werden, von einigen Lichtquellen.



