

## 5. Übungsblatt zur Vorlesung Computergraphik im WS 2022/23

Besprechung am Montag und Dienstag, 05.12. & 06.12.2021

### Aufgabe 1 *BRDF [2 Votierpunkte]*

1. Im Zusammenhang mit Beleuchtungsberechnungen sind BRDFs relevant. Nennen Sie drei Beispiele für Parameter, die in eine BRDF eingehen können.
2. Was besagt der Wert einer BRDF für einen bestimmten Parametersatz?
3. Wie können BRDFs gewonnen werden?
4. Beschreiben Sie den Unterschied zwischen physikalischen und phänomenologischen BRDF-Modellen.

### Aufgabe 2 *Beleuchtung [2 Votierpunkte]*

1. Erklären Sie den Unterschied zwischen spekularer und diffuser Reflexion. Was bedeutet hierbei der Begriff *glossy*?
2. Welche Bedeutung hat die ambiente Komponente bei der Phong Beleuchtung? Kommt diese in der "realen Welt" vor?

### Aufgabe 3 *Beleuchtung im Raytracer [6 Votierpunkte]*

Eine aktualisierte Version des Raytracers aus dem letzten Übungsblatt steht Ihnen als Skelett zur Verfügung. Das Skelett enthält nun zusätzlich die Klasse `Pointlight`, die Punktlichtquellen repräsentiert. In der Szene sind bereits 16 Punktlichter verteilt. Die `Ray`-Klasse wurde um die Member-Variable `depth` ergänzt, die die rekursive Tiefe des Strahls speichert.

**Hinweis:** Im *Debug*-Modus kann die Ausführungszeit des Programms mit Schatten- und Reflexionsstrahlen sehr lange werden. Falls Sie Ihre Implementierung im *Debug*-Modus testen, reduzieren Sie am besten vorübergehend die Anzahl der Lichtquellen in der Szene oder die Auflösung des Ausgabebildes.

1. Implementieren Sie in der Datei `vec3.h` die Funktion `reflect (TODO 1)`, die einen einfallenden Vektor reflektiert (siehe Vorlesung Kapitel 2, Folie 76 als Hinweis).
2. Implementieren Sie in der Datei `main.cpp` die Funktion `computePhongLighting (TODO 2)`. Die Funktion soll das Phong-Beleuchtungsmodell berechnen (siehe Vorlesung Kapitel 2, Folie 84). Verwenden Sie dabei die Lichtfarbe nur bei der Berechnung des spekularen Anteils.

3. Implementieren Sie in der Datei *main.cpp*, in der Funktion `castRay`, die lokale Beleuchtung für getroffene Szeneobjekte (*TODO 3*). Iterieren Sie dafür über alle Lichtquellen. Konstruieren Sie jeweils einen Schattenstrahl, um zu prüfen ob der Schnittpunkt mit dem Objekt von der Lichtquelle beleuchtet wird. Berechnen Sie gegebenenfalls die korrekten Eingabeparameter für die Funktion `computePhongLighting` und rufen Sie die Funktion auf, um die lokale Beleuchtung zu berechnen. Falls der Schnittpunkt im Schatten liegt, *beleuchten* Sie ihn direkt nur mit dem ambienten Term des Phong-Beleuchtungsmodells (dafür benötigen Sie keine extra Funktion).
4. Implementieren Sie in der Datei *main.cpp*, in der Funktion `castRay`, die spiegelnde Reflexion für das getroffenen Szeneobjekte (*TODO 4*). Konstruieren Sie dafür einen Reflexionsstrahl und rufen Sie die Funktion `castRay` rekursiv auf. Denken Sie daran die Rekursionstiefe im Reflexionsstrahl sinnvoll zu setzen, da ihr Programm sonst in eine Endlosschleife gerät!

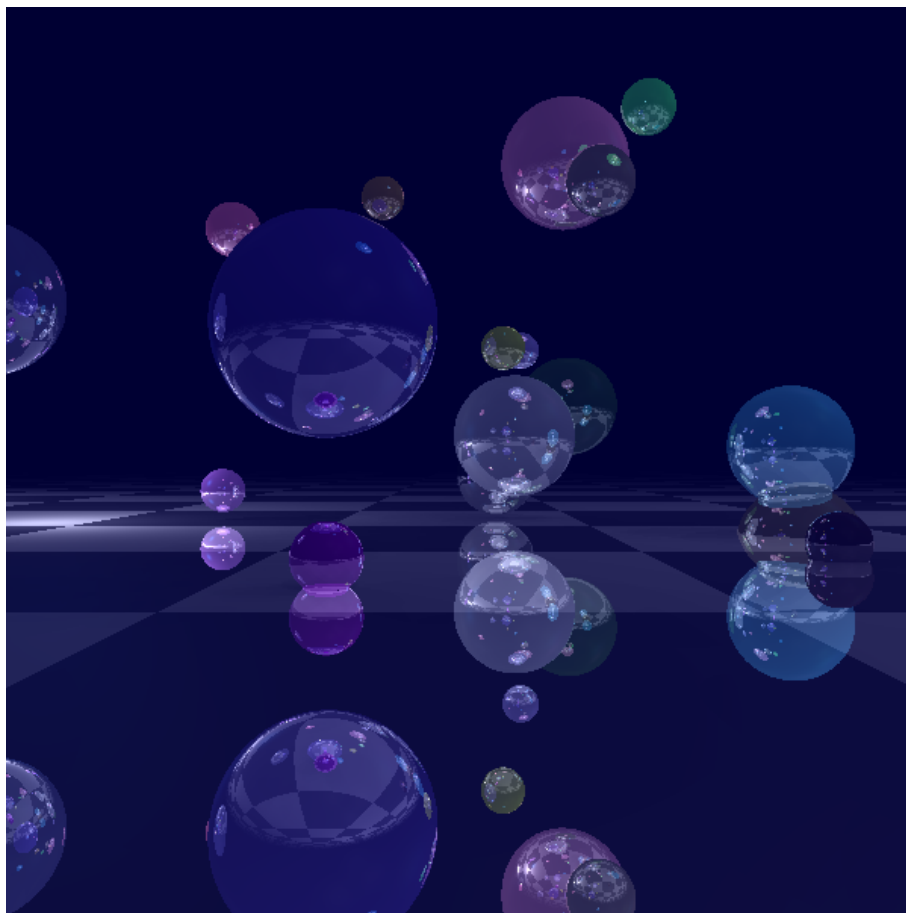


Abbildung 1: Referenzbild der beleuchteten Szene mit Reflexionen bei einer Auflösung von 600x600 Pixeln

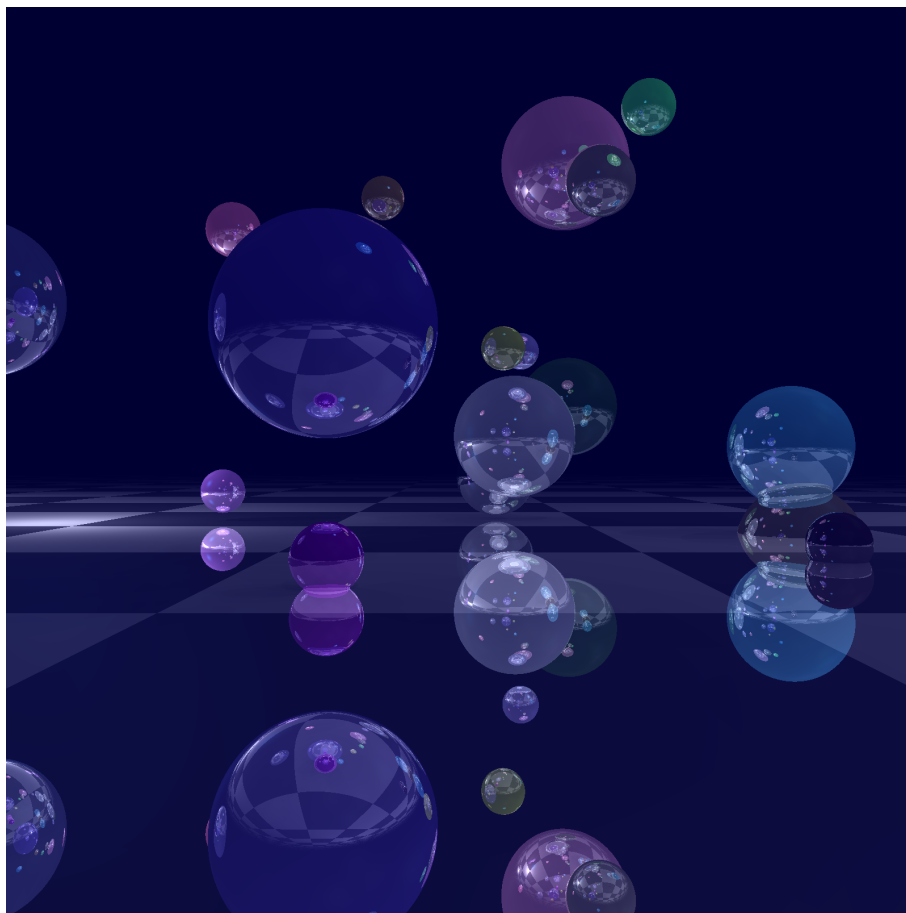


Abbildung 2: Referenzbild der beleuchteten Szene mit Reflexionen bei einer Auflösung von 1600x1600 Pixeln.