

## **Aufgabe 1**

### **Teilaufgabe 1a**

TODO

### **Teilaufgabe 1b**

TODO

### **Teilaufgabe 1c**

TODO

### **Teilaufgabe 1d**

TODO

### **Teilaufgabe 1e**

TODO

### **Teilaufgabe 1f**

TODO

## **Aufgabe 2: Raytracing**

### **Teilaufgabe 2a**

TODO

### **Teilaufgabe 2b**

TODO

### **Teilaufgabe 2c**

TODO

### **Teilaufgabe 2d**

TODO

## **Aufgabe 3**

### **Teilaufgabe 3a**

TODO

### **Teilaufgabe 3b**

TODO

## **Aufgabe 4**

### **Teilaufgabe 4a**

TODO

### **Teilaufgabe 4b**

TODO

## **Aufgabe 5**

TODO

## **Aufgabe 6: Texturierung**

### **Teilaufgabe 6a**

TODO

### **Teilaufgabe 6b**

TODO

### **Teilaufgabe 6c**

TODO

### **Teilaufgabe 6d**

TODO

## **Aufgabe 7: Cube-Maps und Environment-Mapping**

### **Teilaufgabe 7a**

TODO

### **Teilaufgabe 7b**

TODO

## **Aufgabe 8: Hierarchische Datenstrukturen**

### **Teilaufgabe 8a**

TODO

### **Teilaufgabe 8b**

TODO

### Teilaufgabe 8c

TODO

### Teilaufgabe 8d

TODO

## Aufgabe 9: Rasterisierung und OpenGL

TODO

## Aufgabe 10: Tiefenpuffer und Transparenz

### Teilaufgabe 10a

TODO

### Teilaufgabe 10b

TODO

### Teilaufgabe 10c

TODO

## Aufgabe 11: Phong-Shading und Phong-Beleuchtungsmodell

```
shader.vert
1 uniform mat4 matN; // Normalentransformation (Objekt -> Kamera)
2 uniform mat4 matM; // Modelltransformation
3 uniform mat4 matV; // Kameratransformation
4 uniform mat4 matP; // Projektionstransformation
5 uniform mat4 matMV; // Model-View-Matrix
6 uniform mat4 matMVP; // Model-View-Projection-Matrix
7
8 in vec3 P; // Eingabe-Vertex in Objektkoordinaten
9 in vec3 n; // Eingabenormale in Objektkoordinaten
```

```
10
11 out vec3 P_k; // Vertex-Position in Kamerakoordinaten
12 out vec3 n_k; // Vertex-Normale in Kamerakoordinaten
13
14 void main() {
15     // P_k = TODO;
16     // n_k = TODO;
17     // gl_Position = TODO;
18 }
```

---