

Aufgabe 1: Raytracing

Teilaufgabe 1a

TODO

Teilaufgabe 1b

TODO

Aufgabe 2

Teilaufgabe 2a

Abbildung 1: Whatever

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

Teilaufgabe 5a

TODO

Teilaufgabe 5a

TODO

Aufgabe 6

Teilaufgabe 6a

TODO

Teilaufgabe 6b

TODO

Teilaufgabe 6c

```
1  in vec3 A; // Ursprung des Strahls.
2  in vec3 D; // Die normalisierte Richtung des Strahls.
3  in float tMax; // Abbruchkriterium: maximale Suchdistanz.
4  uniform float epsilon; // Toleranz
5
6  // Distanzfunktion. Liefert den Abstand von x zur nächsten Fläche.
7  float DF( vec3 x ) { ... }
8
9  // Implementieren Sie Sphere Tracing in dieser Funktion.
10 bool sphereTrace( out vec3 pos, out int steps ) {
11     pos = A;
12     steps = 0;
13     float t = 0.;
14     while (t < tMax) {
15         float d = DF(pos);
16         pos += d * D;
17         if (abs(d) < epsilon) {
18             return true;
19         }
20     }
21     return false;
22 }
```

Aufgabe 7

Teilaufgabe 7a

TODO

Teilaufgabe 7b

TODO

Aufgabe 8

Teilaufgabe 8a

TODO

Teilaufgabe 8b

TODO

Teilaufgabe 8c

TODO

Aufgabe 9

```

1  _____ keyframing.vert _____
2  in vec4 p; // Position des Vertex in Objektkoordinaten.
3  uniform float t; // Aktueller Zeitpunkt.
4  uniform float t1; // Die Zeitpunkte der drei Keyframes.
5  uniform float t2;
6  uniform float t3;
7  uniform mat4 M1; // Die drei Transformationsmatrizen (Objekt->Welt).
8  uniform mat4 M2;
9  uniform mat4 M3;
10 uniform mat4 VP; // Die View-Projection-Matrix.
11
12 void main() {
13     vec4 pWorld;
14     if (t < t2) {
15         pWorld = mix(M1 * p, M2 * p, (t - t1) / (t2 - t1));
16     } else {
17         pWorld = mix(M2 * p, M3 * p, (t - t2) / (t3 - t2));
18     }
19 }
```

```
19     gl_Position = VP * pWorld;  
20 }
```

Aufgabe 10

Teilaufgabe 10a

```
                                shader.frag  
1 void renderScene() {  
2     // Setup vor dem Löschen von Frame- und Tiefenpuffer  
3     glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );  
4     // Zeichnen der Szene ab hier  
5  
6     //TODO  
7 }
```

Teilaufgabe 10b

TODO

Teilaufgabe 10c

TODO

Aufgabe 11: Bézierkurven

Teilaufgabe 11a

TODO

Teilaufgabe 11b

TODO