

# Interaktive Computergrafik



**Prof. Dr. Frank Steinicke**  
Human-Computer Interaction  
Department of Computer Science  
University of Hamburg



# Interaktive Computergrafik

## Übung 9

**Christoph Brauer, Susanne Schmidt**

Human-Computer Interaction, University of Hamburg

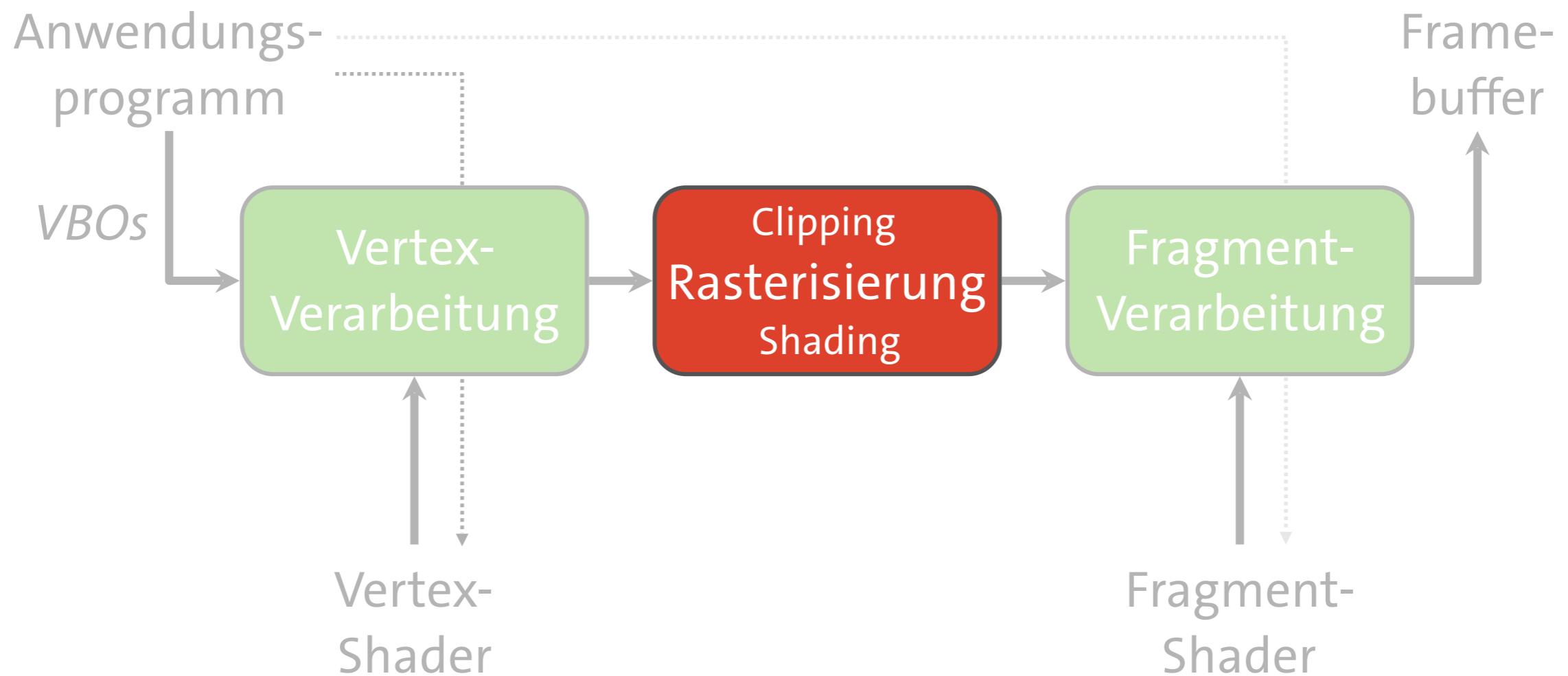


# Interaktive Computergrafik

## Übung 9

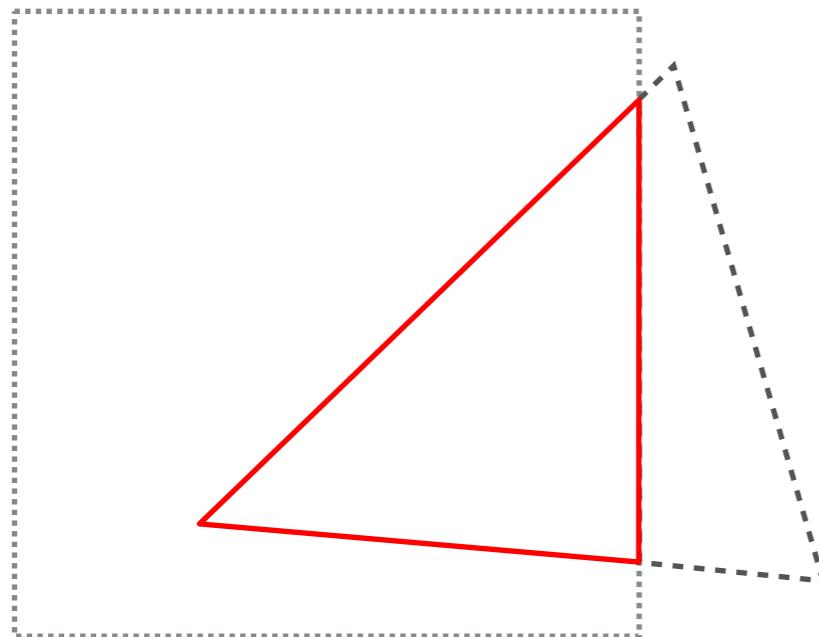
### Shading

# Einordnung



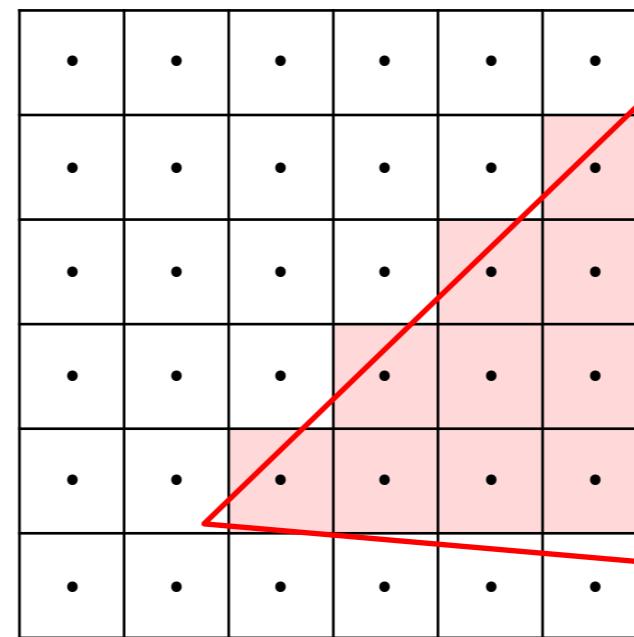
# Aufgaben

## 1. Clipping



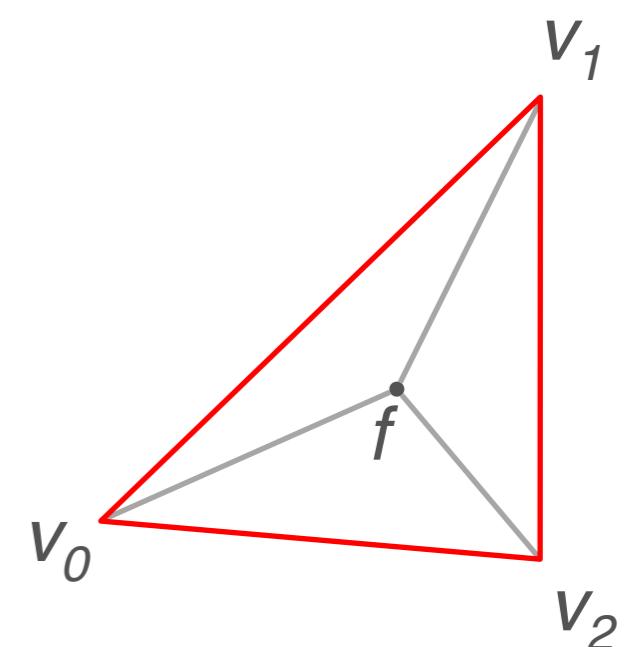
Abschneiden  
nicht-sichtbarer  
Polygonteile

## 2. Rasterisierung



Bestimmung  
abgedeckter  
Pixel

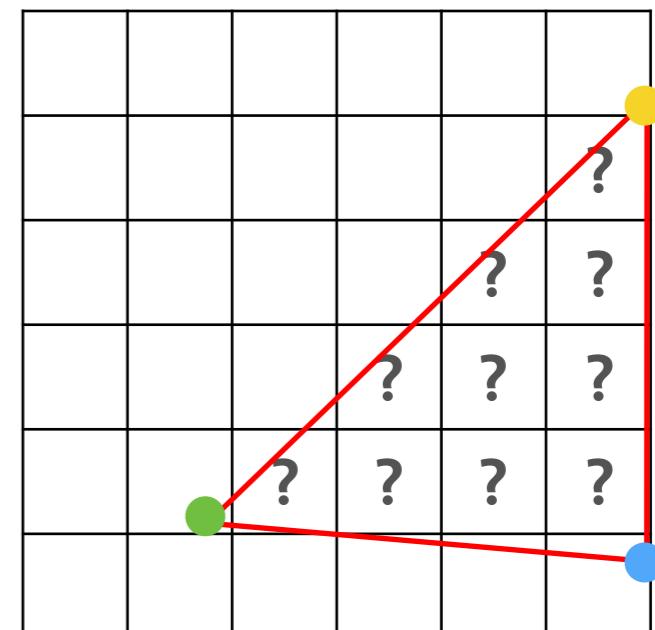
## 3. Shading



Interpolation der  
Fragmentattribute  
aus Eckpunkten

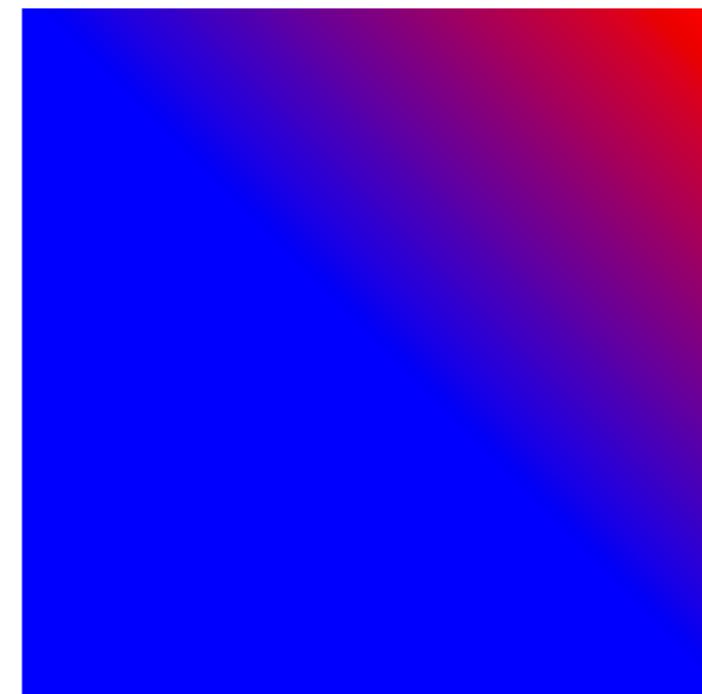
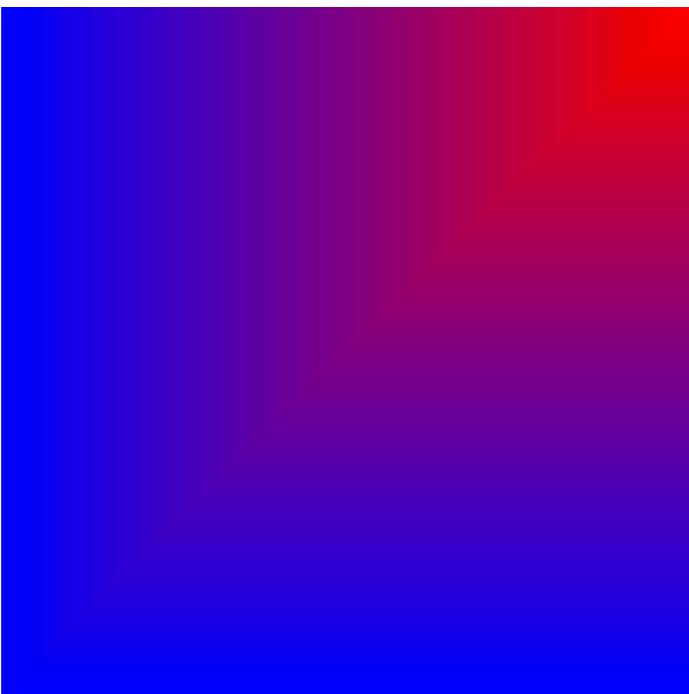
# Interpolation

- Ausgabe des Vertex-Shaders: Attribute pro Eckpunkt (z.B. Farbe, Normale)
- **(Lineare) Interpolation:**  
Berechnung der Attributwerte für einzelne Fragmente durch gewichtete Summierung der Eckpunkt-Attribute



# Interpolation

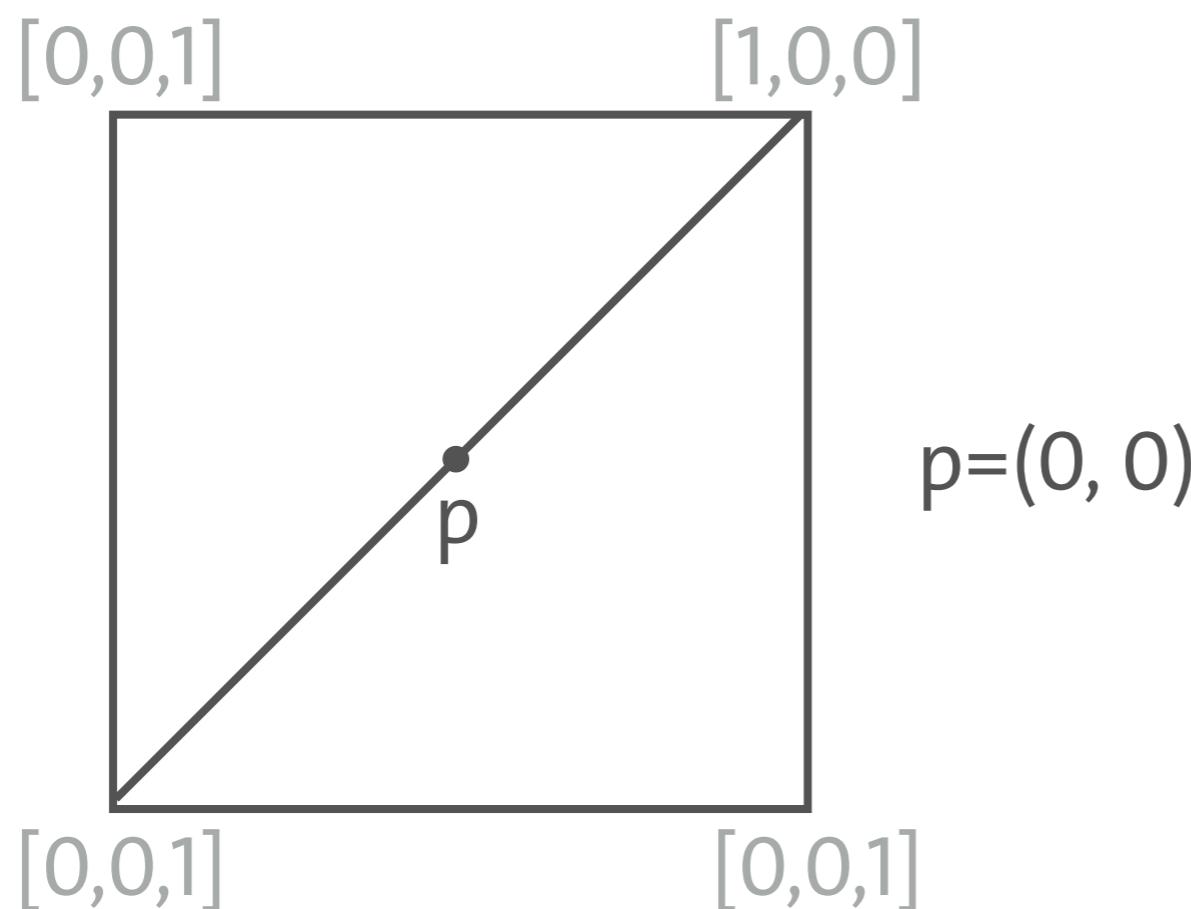
## Beispiel



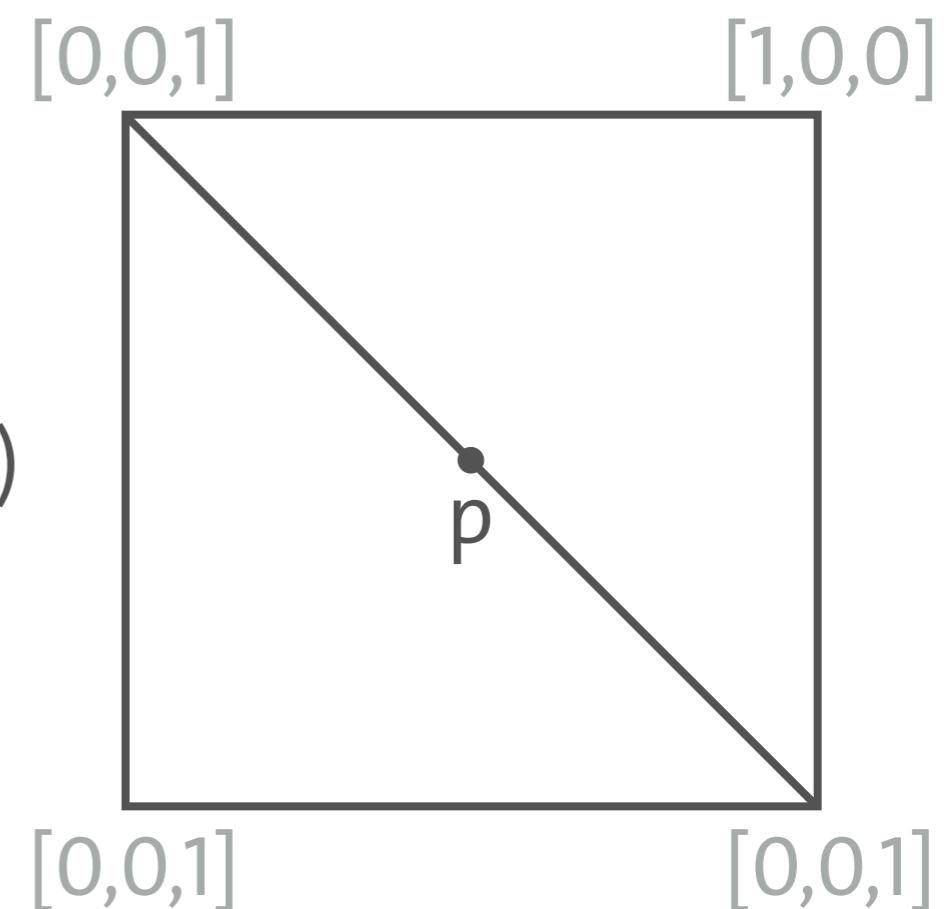
→ Ergebnisse abhängig von Triangulation

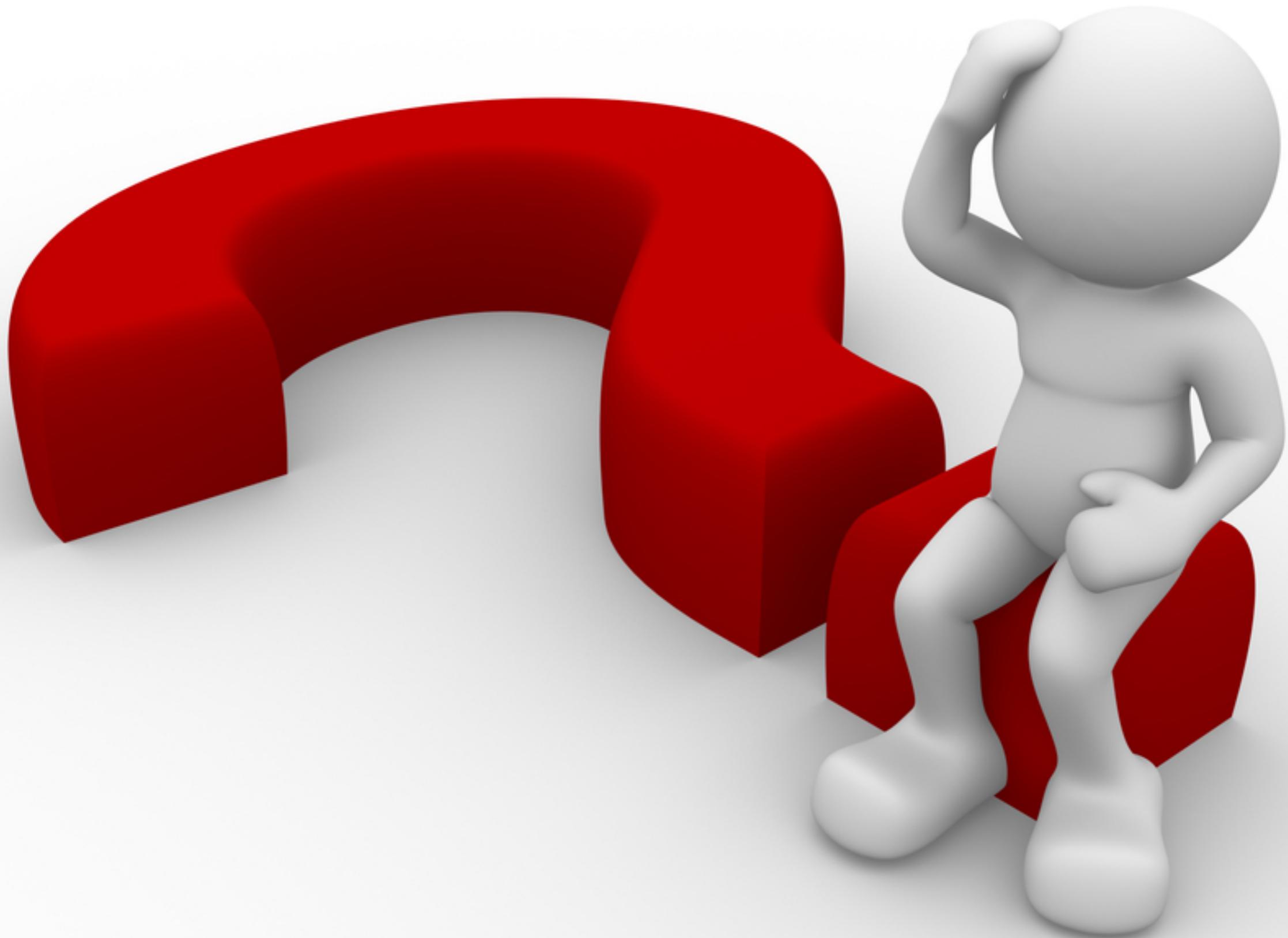
# Interpolation

## Beispiel



$$p=(0, 0)$$







# Interaktive Computergrafik

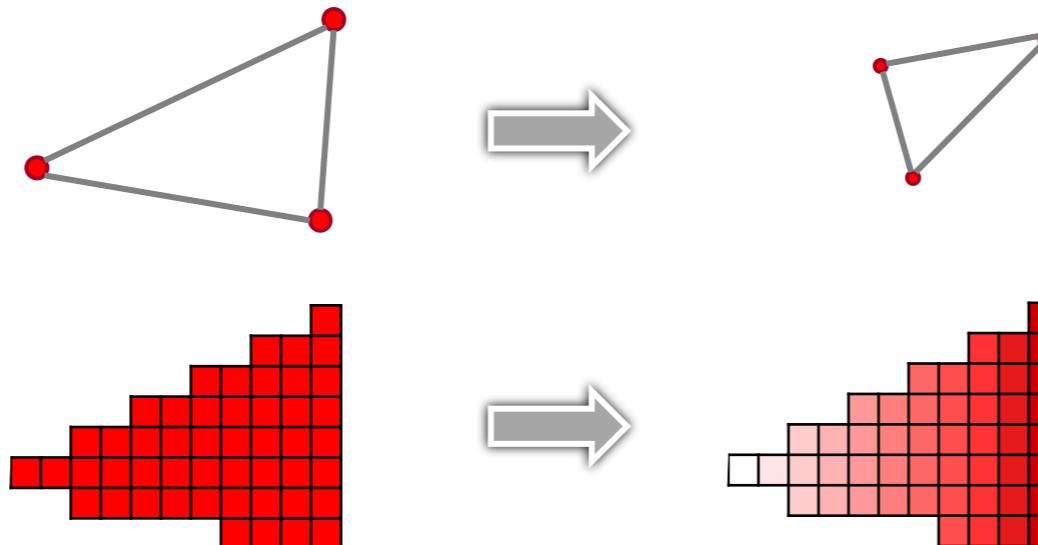
## Übung 9

Fragment Shader

# Wiederholung: Shader

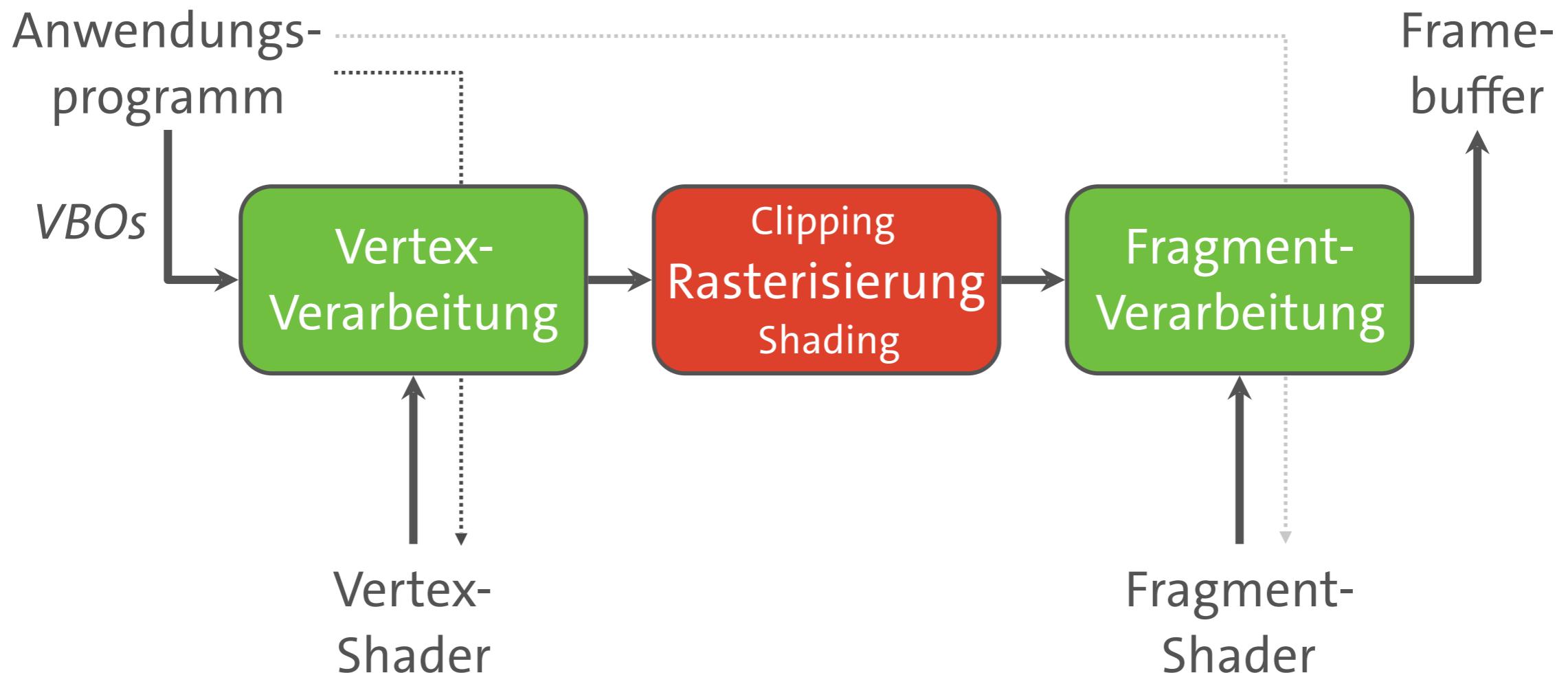
## Allgemein

- **Shading** = Veränderung einzelner Vertices bzw. Fragmente innerhalb der Grafikpipeline



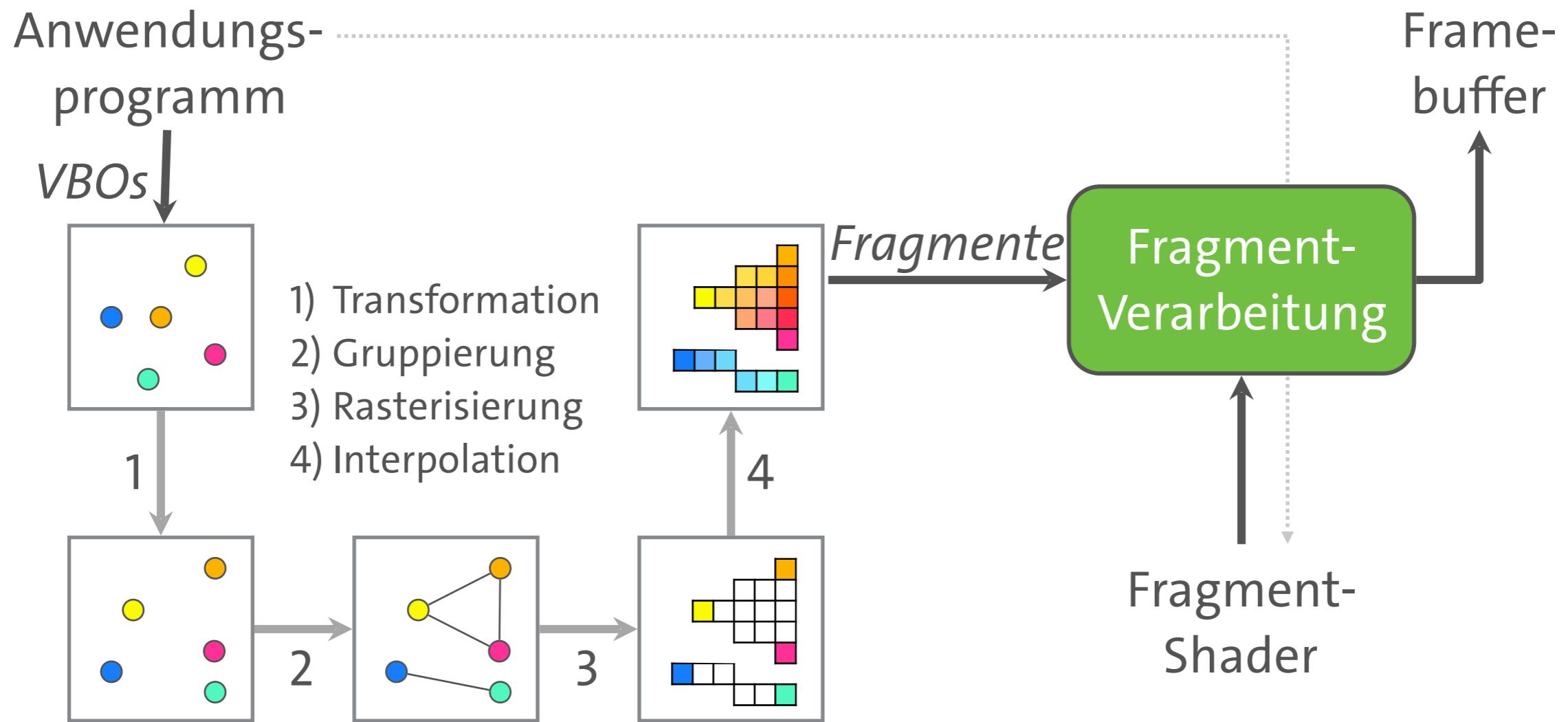
# Fragment Shader

## Einordnung



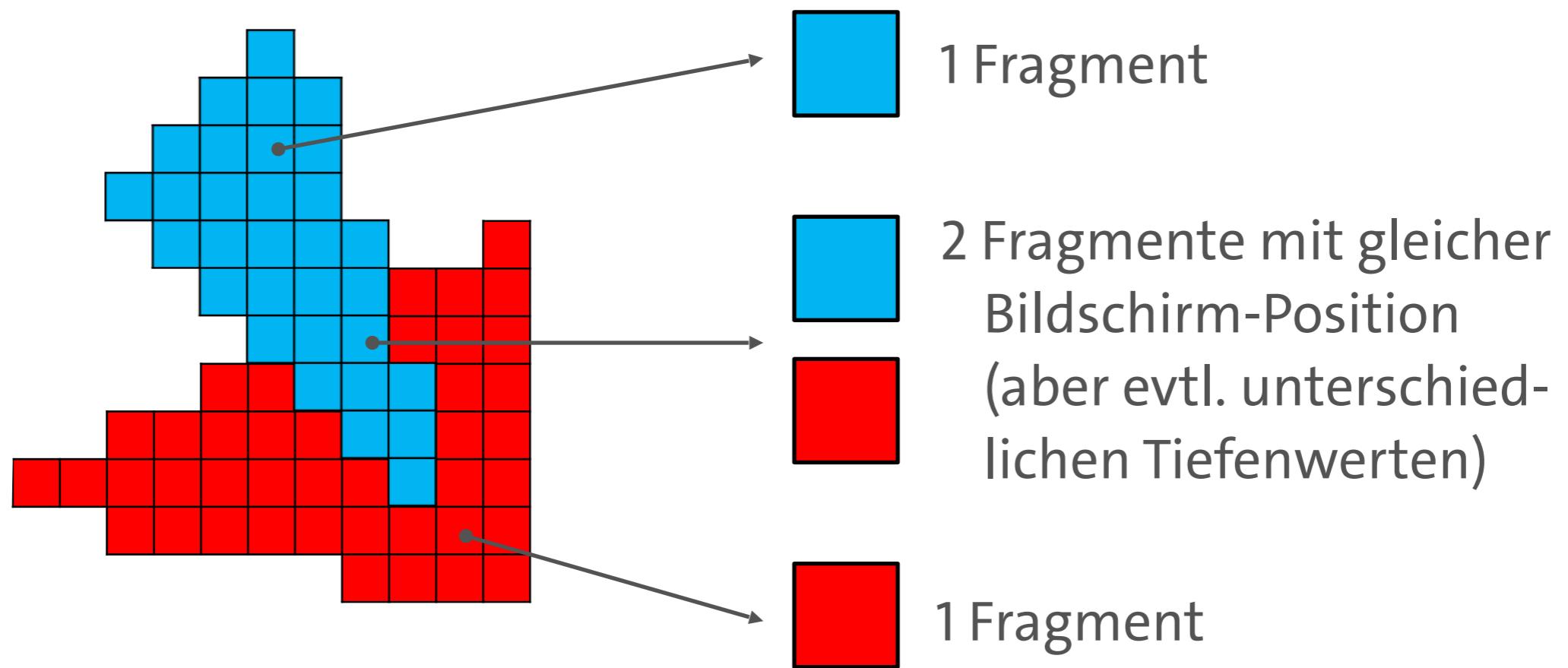
# Fragment Shader

## Einordnung



# Was ist ein Fragment?

- „Vorstufe eines Pixels“



# Fragment-Shader

## Aufgaben

Interpolierte Informationen  
(z.B. Farbe, Normale)



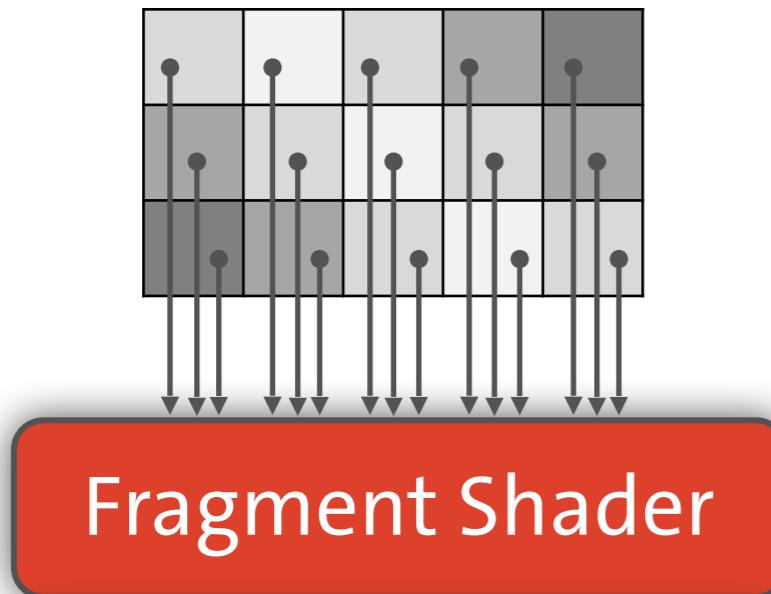
Fragment Shader

- Beleuchtung
- Texturierung
- Nebel



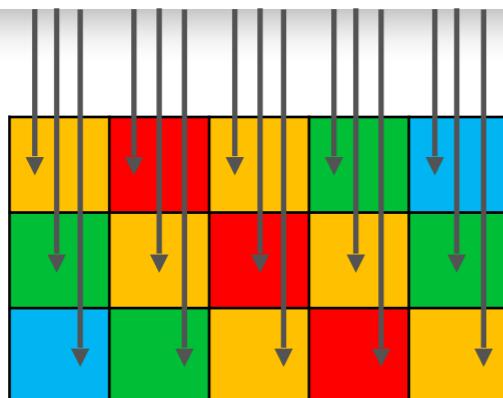
Farbwert pro Fragment

# Fragment Shader



- Beleuchtung
- Texturierung
- Nebel

- Ausführung des GLSL-Shaders für jedes einzelne Fragment
- **parallele Verarbeitung**  
→ kein Zugriff auf andere Fragmente innerhalb des Shaders





Szene ohne Texturen (The Last of Us)



Szene mit Texturen (The Last of Us)



Wasser mit niedrigem Detailgrad (Watch Dogs)



Wasser mit hohem Detailgrad (Watch Dogs)



Szene ohne Toon Shader (Far Cry)



Szene mit Toon Shader (Far Cry)



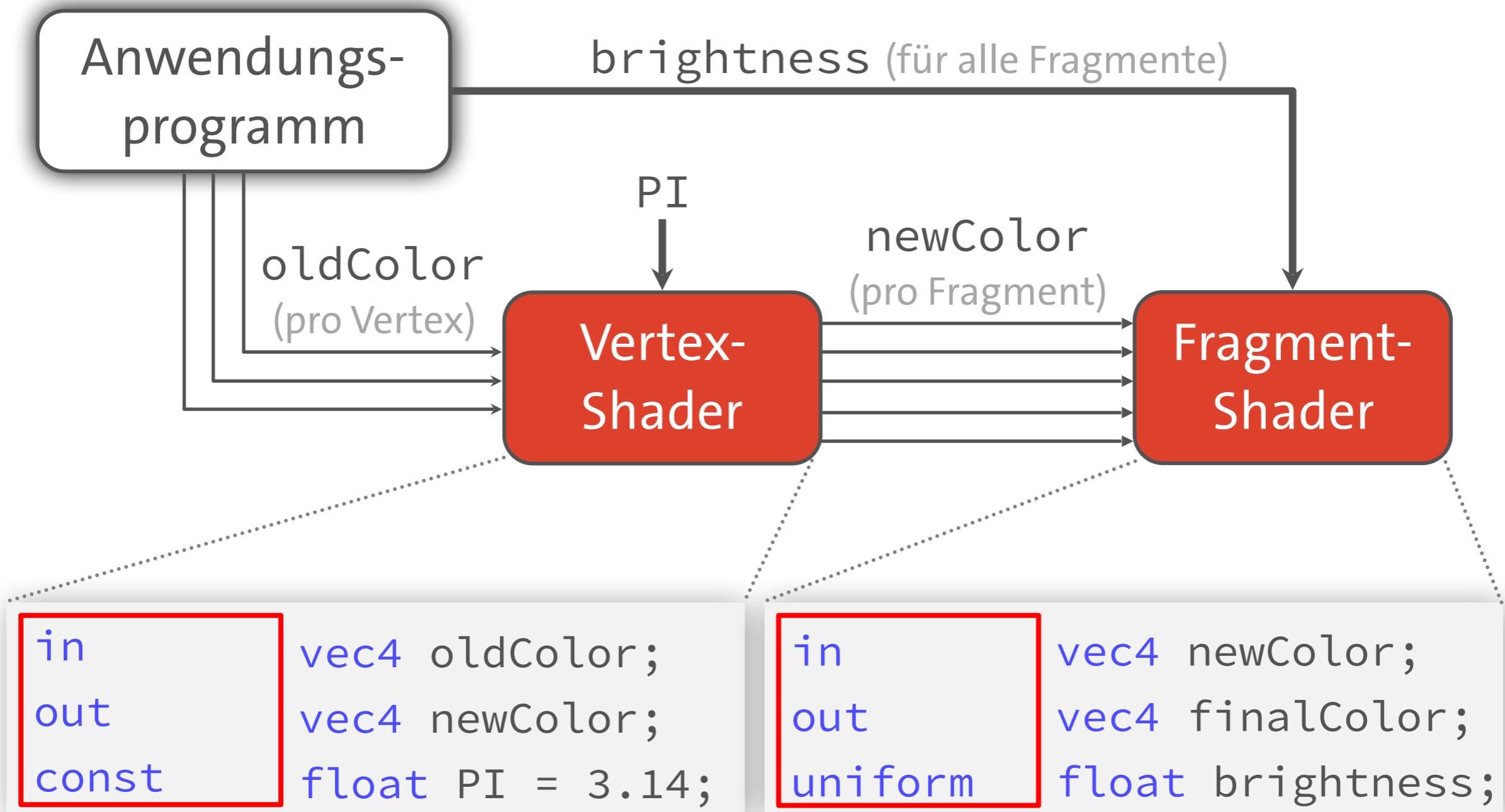
Volumetrischer Nebel (Assassin's Creed Unity)

# Beispielprogramm #1

```
precision mediump float; ← Genaugkeit  
in vec4 vfColor;  
out vec4 fColor;  
  
void main() {  
    fColor = vfColor;  
}
```

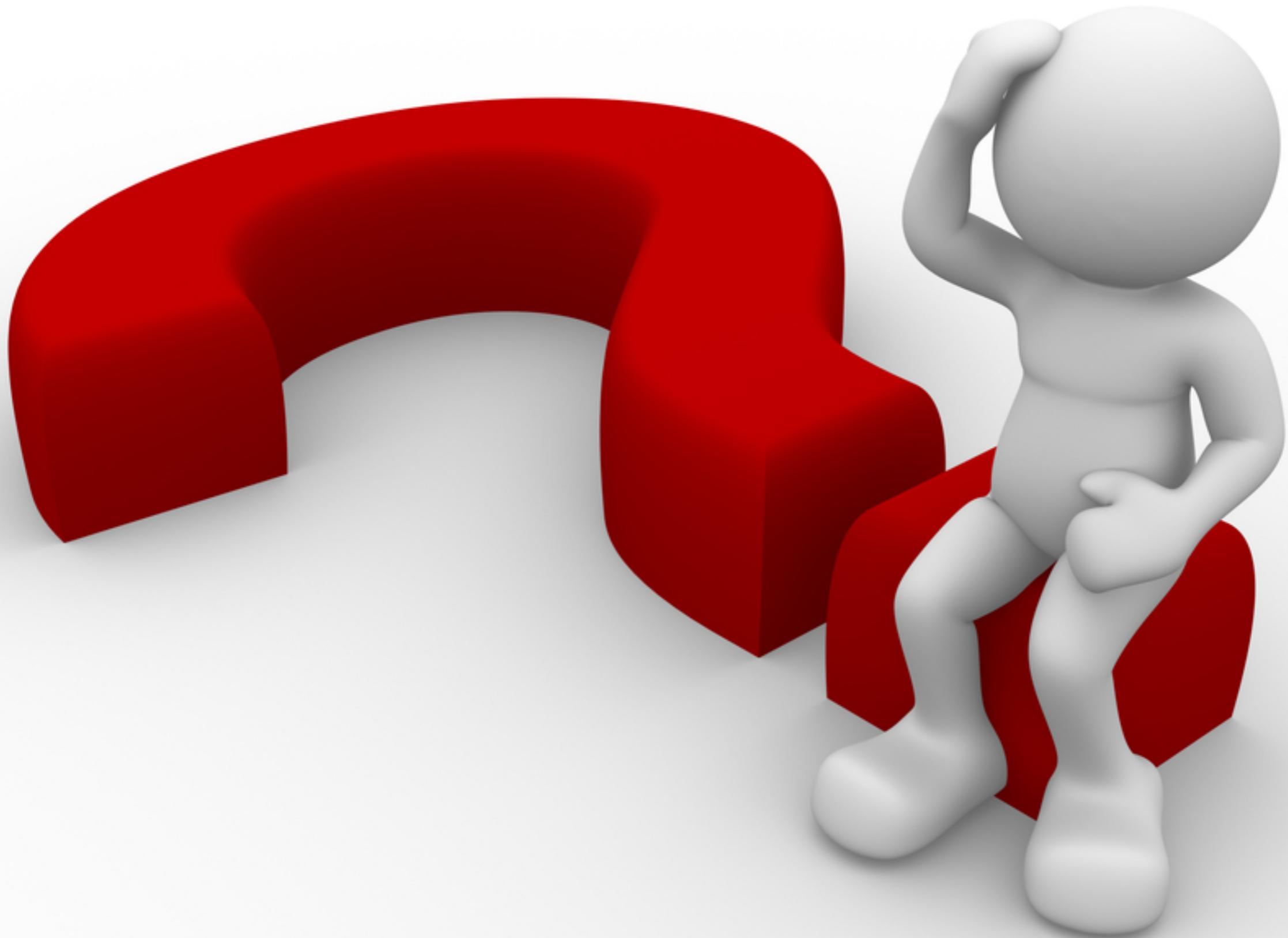
# Typenqualifizierer

## Beispiel



# Beispielprogramm #2

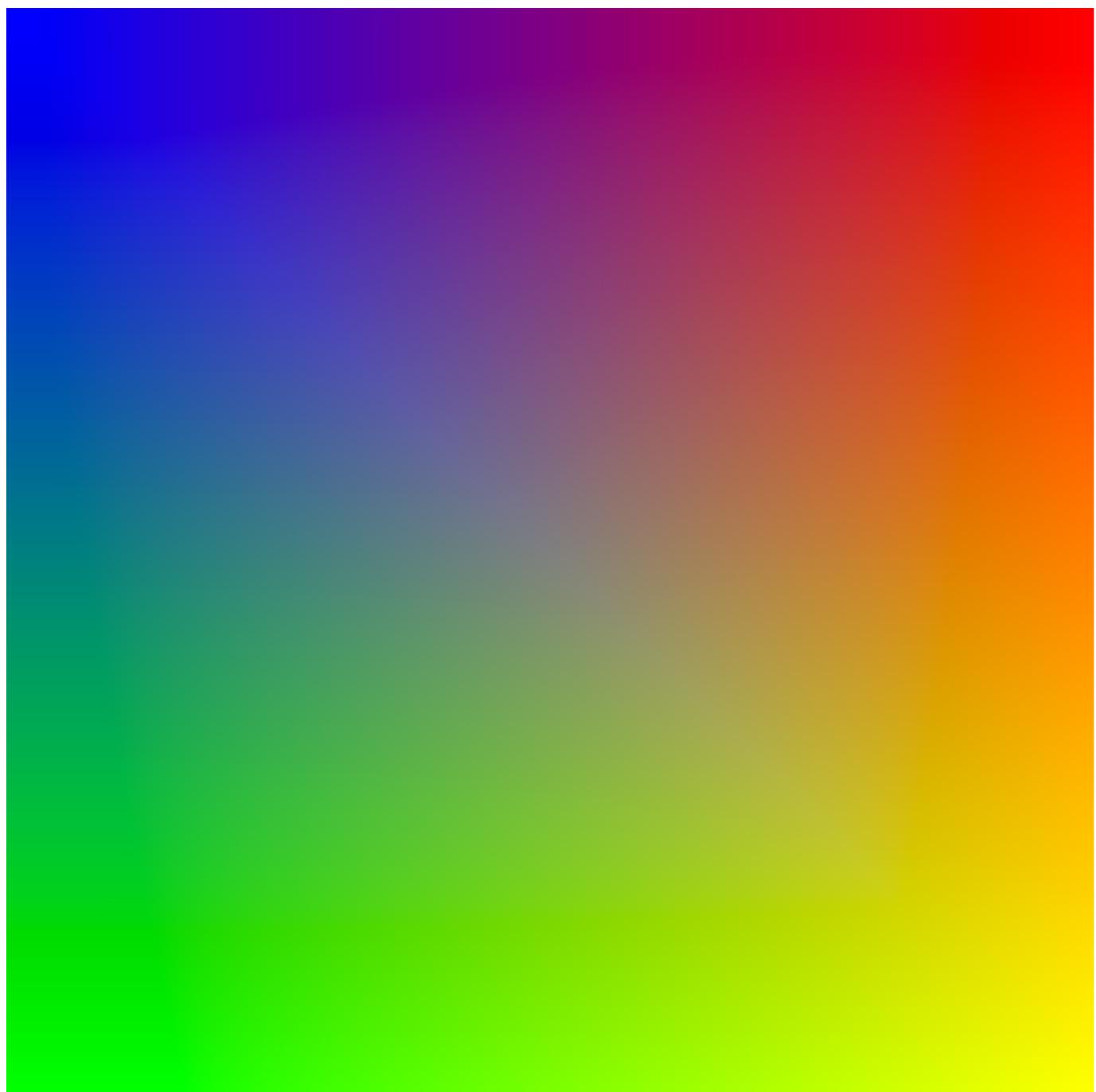
```
precision mediump float;  
  
in vec4 vfColor;  
out vec4 fColor;  
  
void main() {  
    float y = 0.30 * vfColor.r  
        + 0.59 * vfColor.g  
        + 0.11 * vfColor.b;  
    fColor = vec4(y, y, y, 1);  
}
```



# Gruppenarbeit



Zeichnen Sie das folgende canvasfüllende Quad (im Anwendungsprogramm).



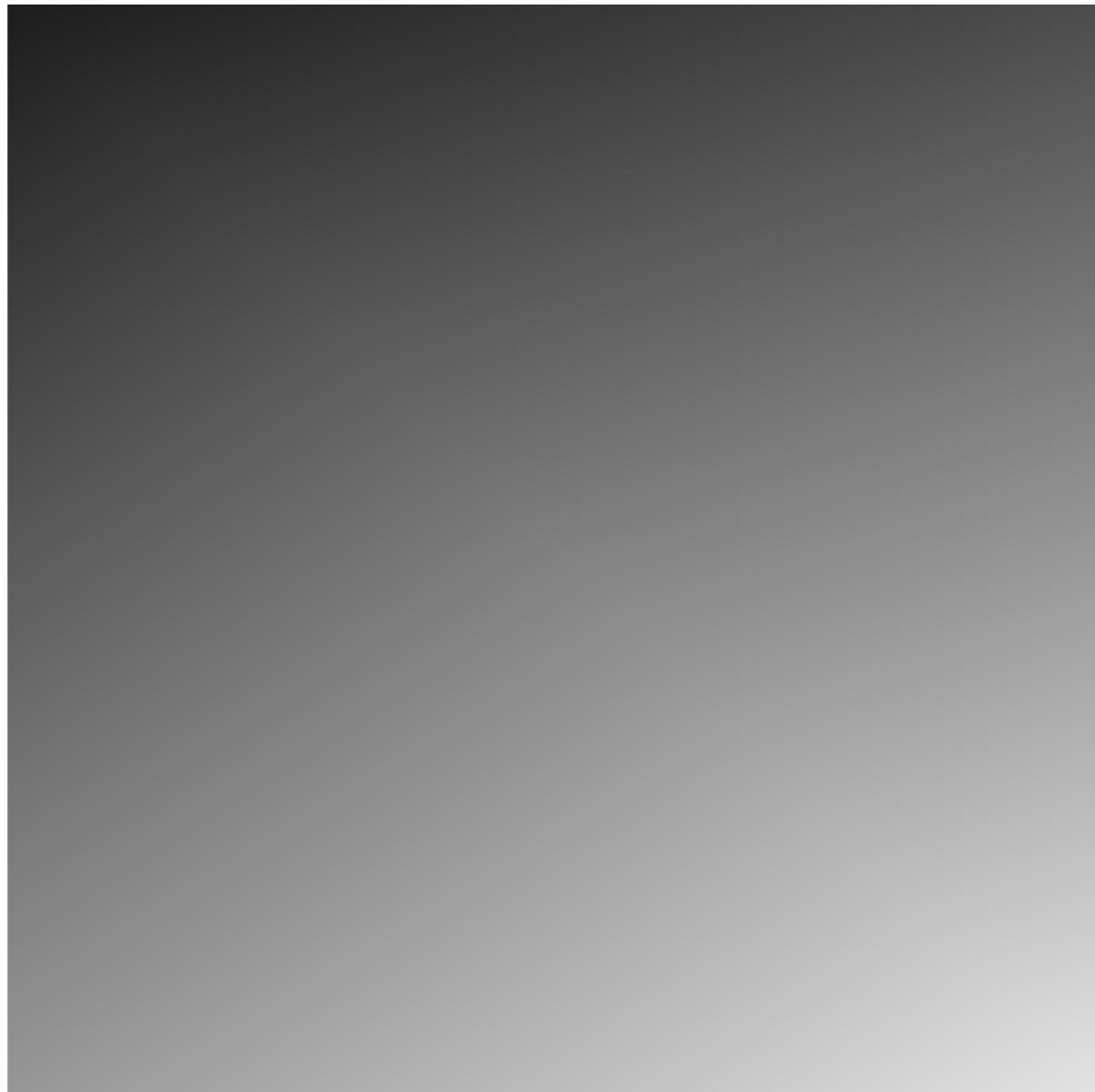
# Gruppenarbeit



Schreiben Sie einen Fragment-Shader, der ...



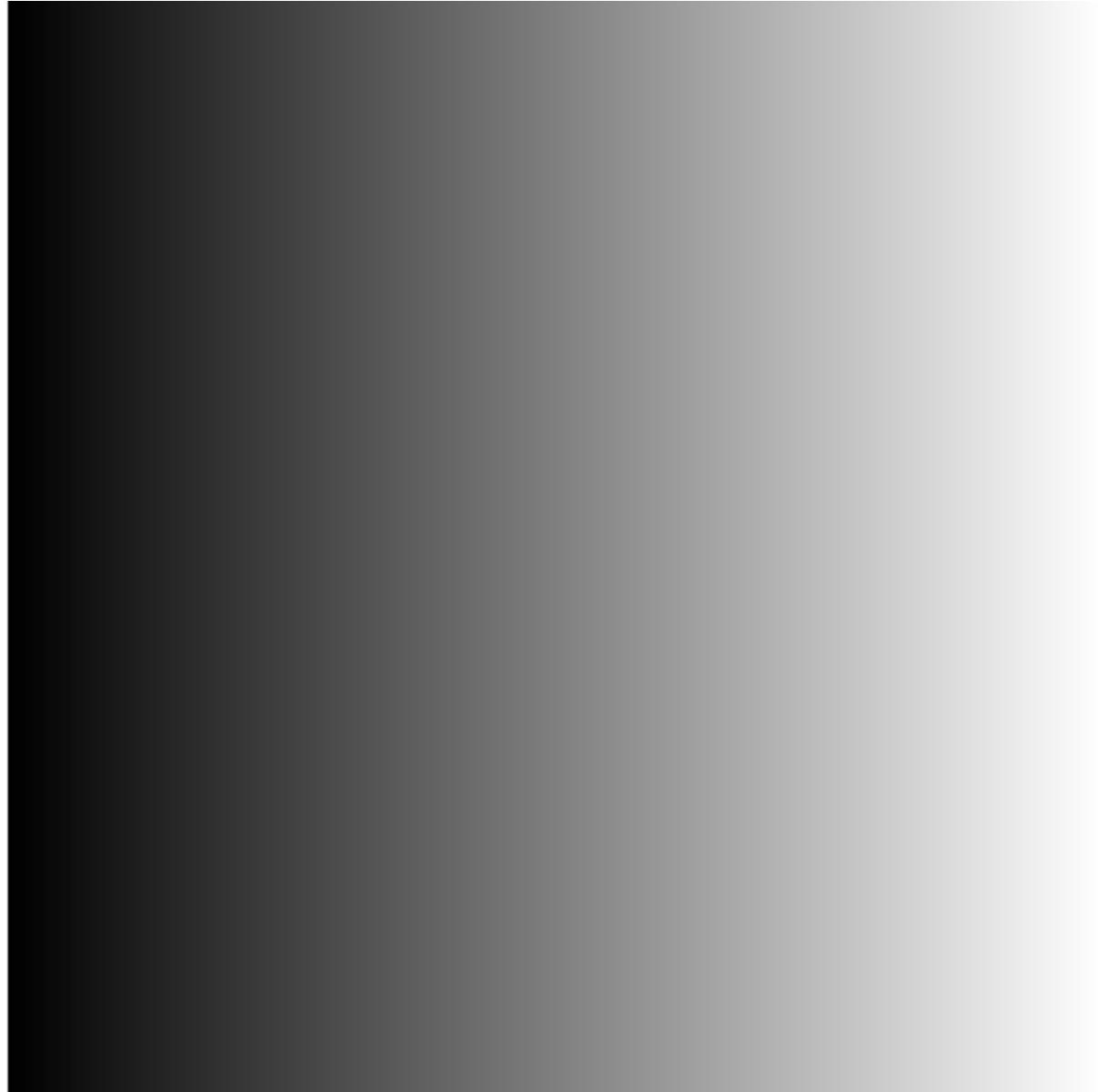
... das gesamte Quad rot färbt.



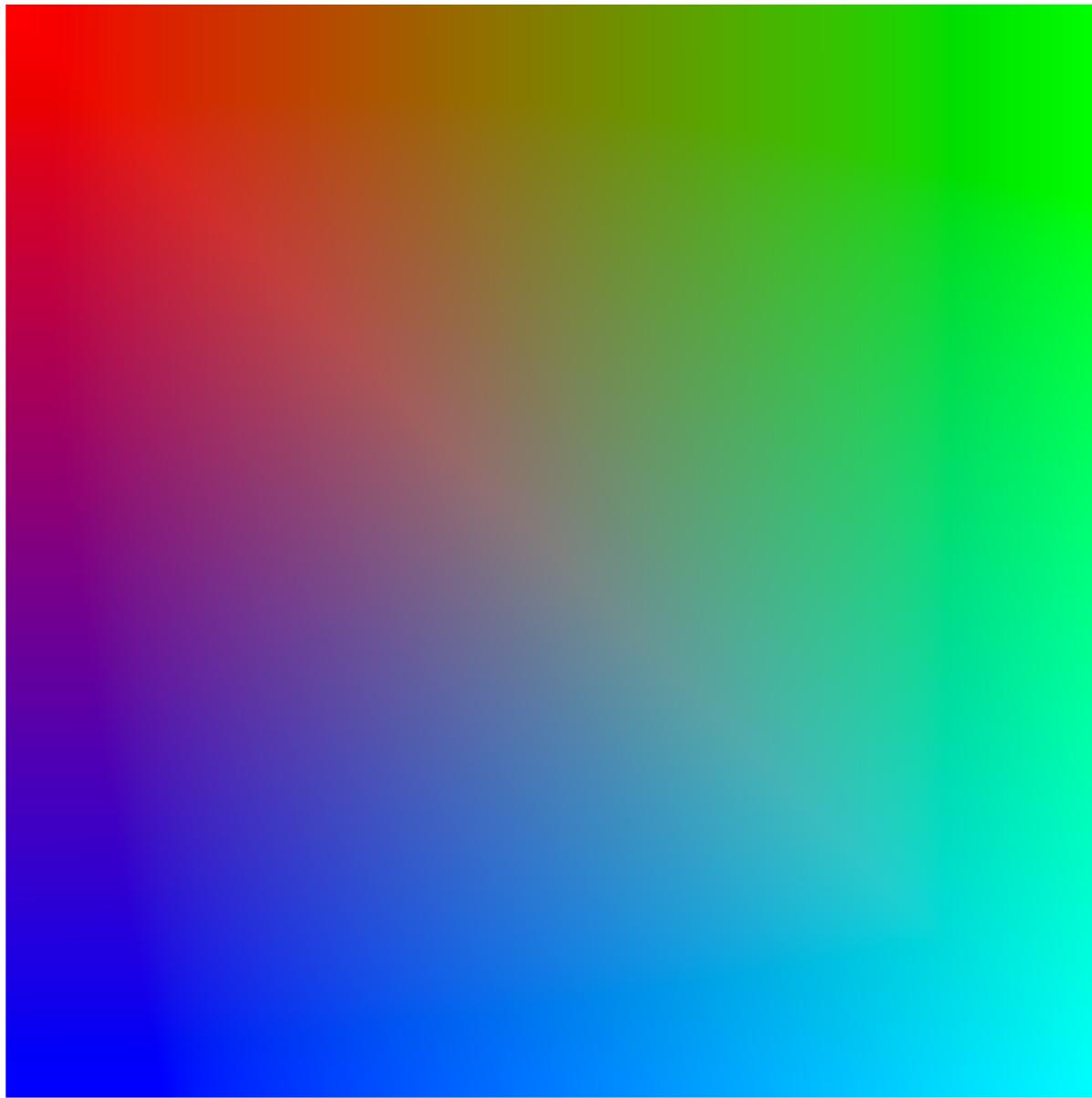
... das Bild in Graustufen darstellt.



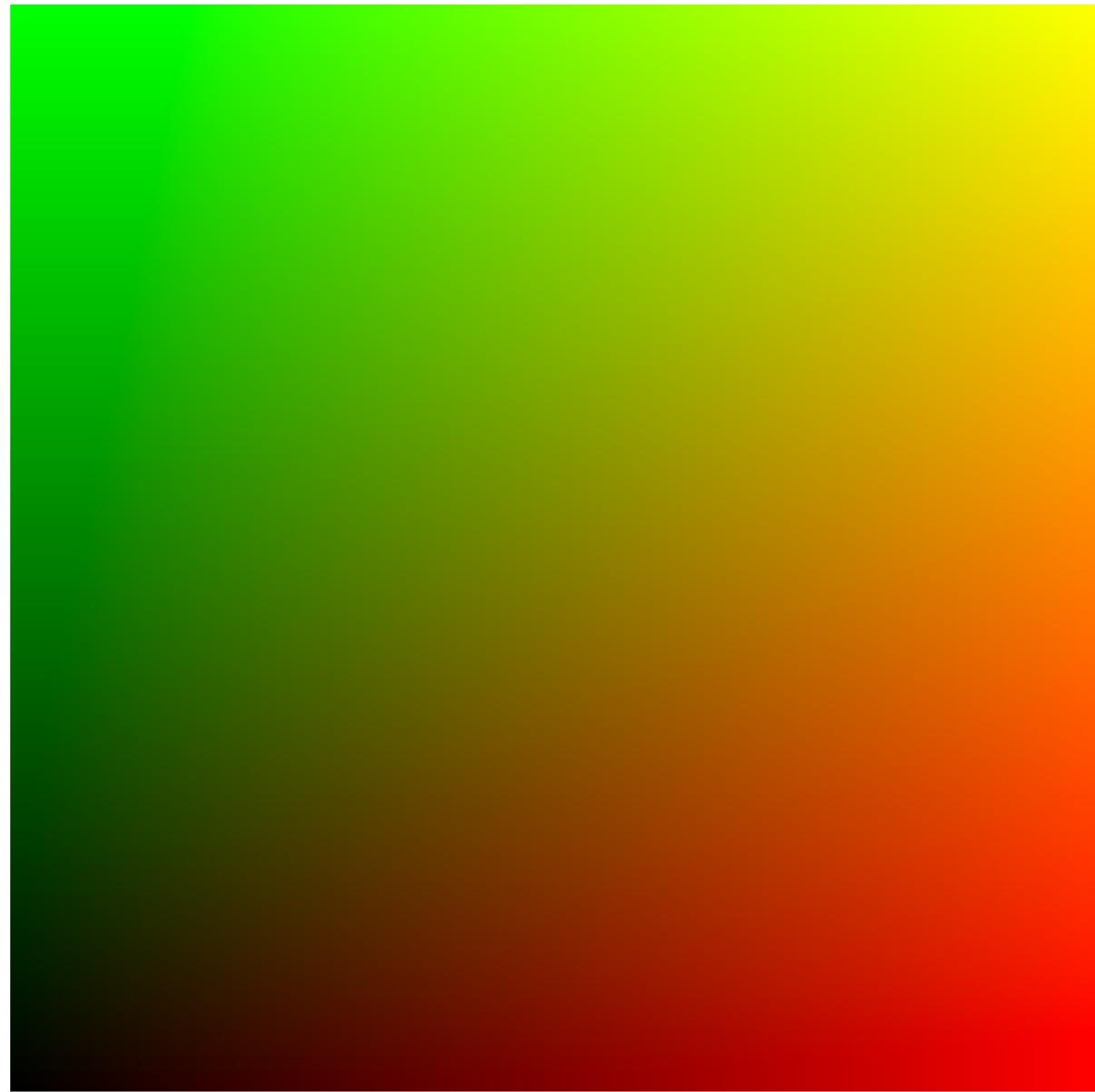
... den Blau-Kanal in Graustufen darstellt.



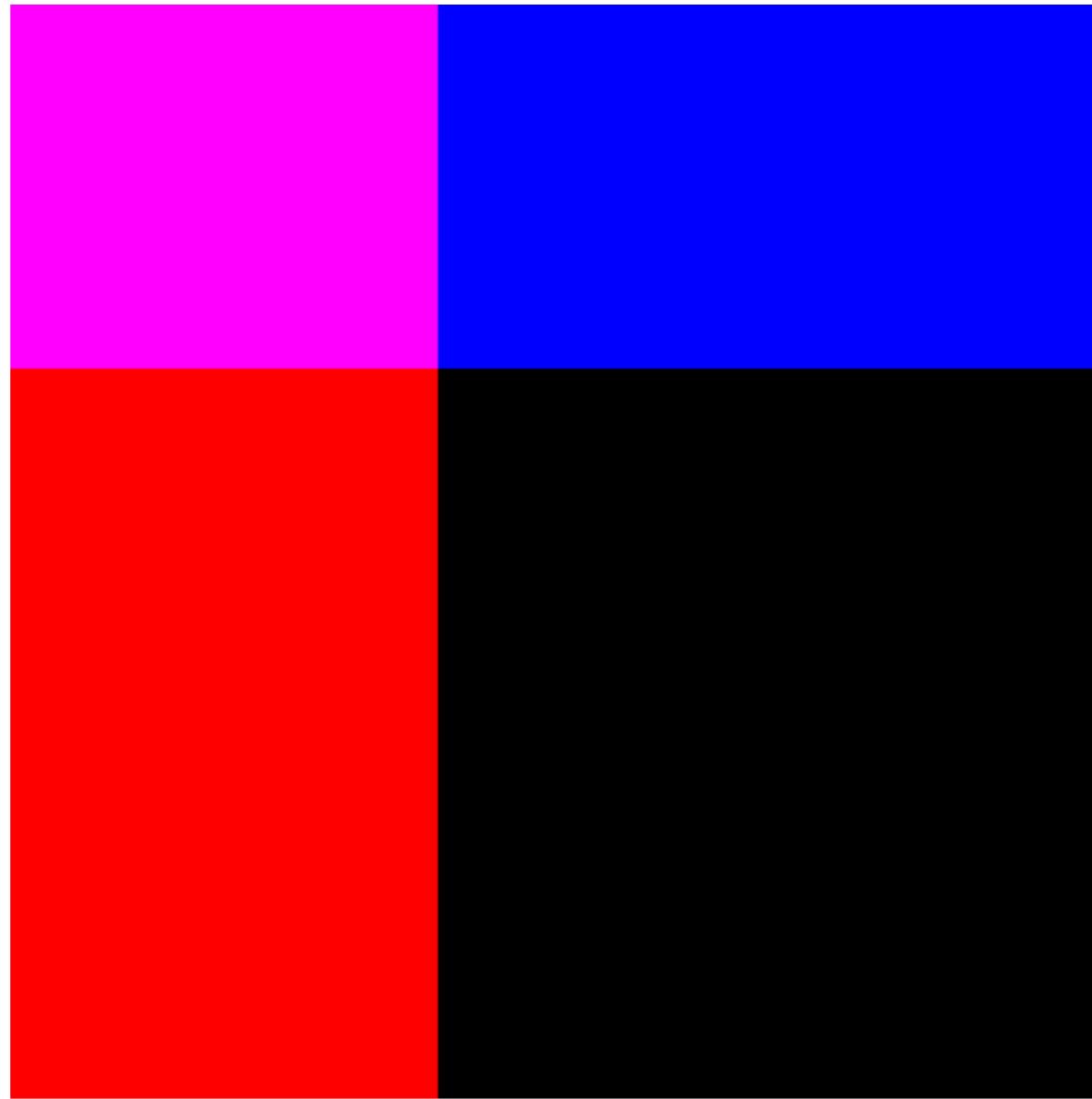
... den Rot-Kanal in Graustufen darstellt.



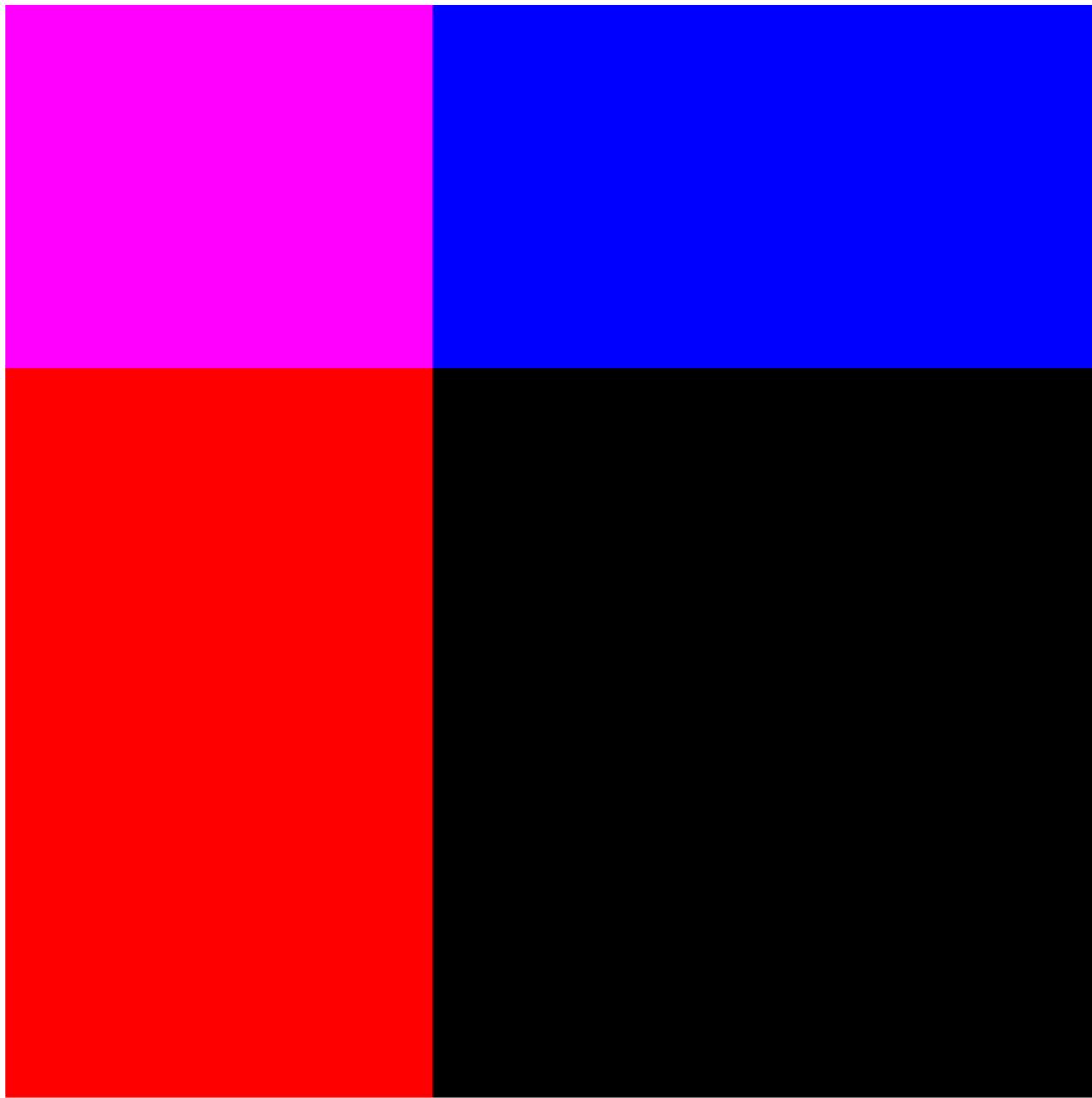
... alles Rote durch Blau, alles Grüne durch  
Rot und alles Blaue durch Grün ersetzt.



... den Rot-Kanal abhängig von der x-Position und den Grün-Kanal abhängig von der y-Position des Fragments bestimmt.



... eine konstante Koordinate als Mittelpunkt setzt und die Quartale wie dargestellt färbt.



... beim Mausklick die Position der Maus als  
Mittelpunkt setzt und die Quartale wie  
dargestellt färbt.