

# Aufbau einer modernen Rendering-Pipeline

## Von der Anwendung zum Bildschirm

Fakultät für Informatik - Lehrstuhl für Computergrafik - Institut für Visualisierung und Datenanalyse



# Motivation



Motivation

Anwendung

Geometry Stage

Rasterisierung

Per Frag Ops

Koordinatensysteme

Compute Shader

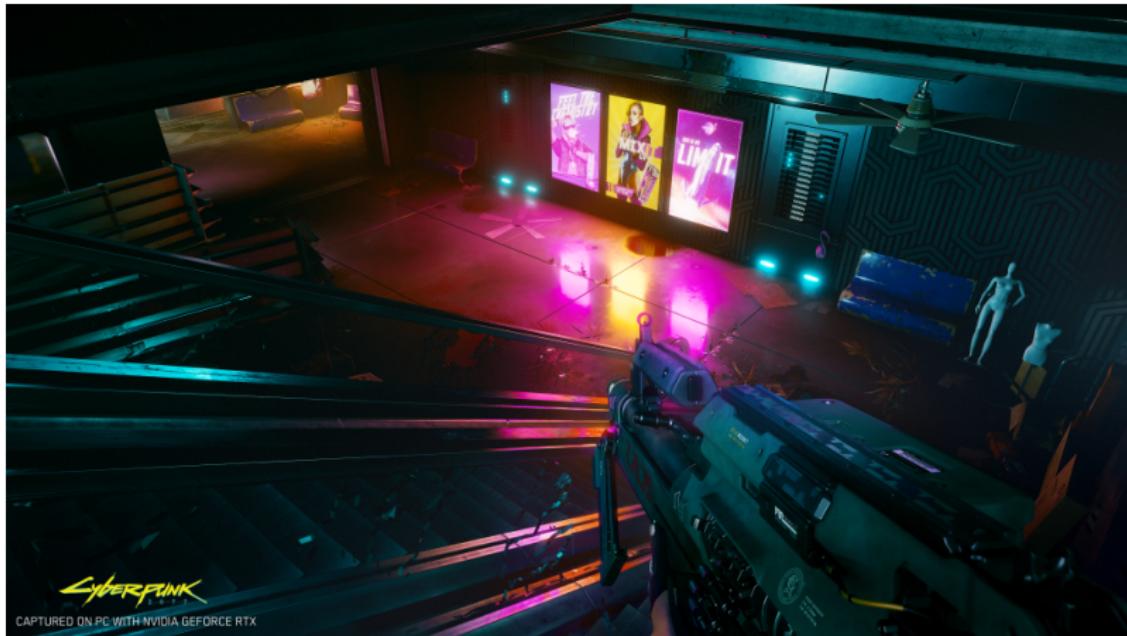
Ausblick

Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

23. Juni 2019

◀ ▶ ⌂ ⌃ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ⌊ ⌋ ⌁ ⌂ ⌃ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ⌊ ⌋ ⌁

# Motivation



Motivation



Anwendung



Geometry Stage



Rasterisierung



Per Frag Ops



Koordinatensysteme



Compute Shader



Ausblick

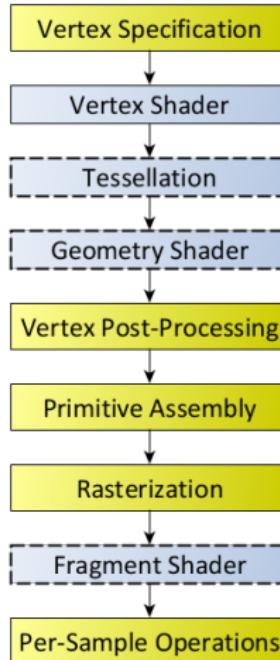


Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

23. Juni 2019



# Motivation



7

Motivation  
○●

Anwendung  
○

Geometry Stage  
○○○○○○○○

Rasterisierung  
○

Per Frag Ops  
○

Koordinatensysteme  
○

Compute Shader  
○

Ausblick  
○○○○○○○○

Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

23. Juni 2019



# Anwendung

```
1 # Blender v2.79 (sub 0) OBJ File: ''
2 # www.blender.org
3 mtllib Suzanne.mtl
4 o Suzanne
5 v 0.437500 0.164062 0.765625
6 v -0.437500 0.164062 0.765625
7 v 0.500000 0.093750 0.687500
8 v -0.500000 0.093750 0.687500
9 v 0.546875 0.054688 0.578125
```

```
4 newmtl Material
5 Ns 96.078431
6 Ka 1.000000 1.000000 1.000000
7 Kd 0.640000 0.640000 0.640000
8 Ks 0.500000 0.500000 0.500000
9 Ke 0.000000 0.000000 0.000000
10 Ni 1.000000
11 d 1.000000
12 illum 2
13
```

Abbildung: Beispielhafte Objektdefinition

Abbildung: Beispielhafte Materialdefinition

Motivation  
○○

Anwendung  
●

Geometry Stage  
○○○○○○○○

Rasterisierung  
○

Per Frag Ops  
○

Koordinatensysteme  
○

Compute Shader  
○

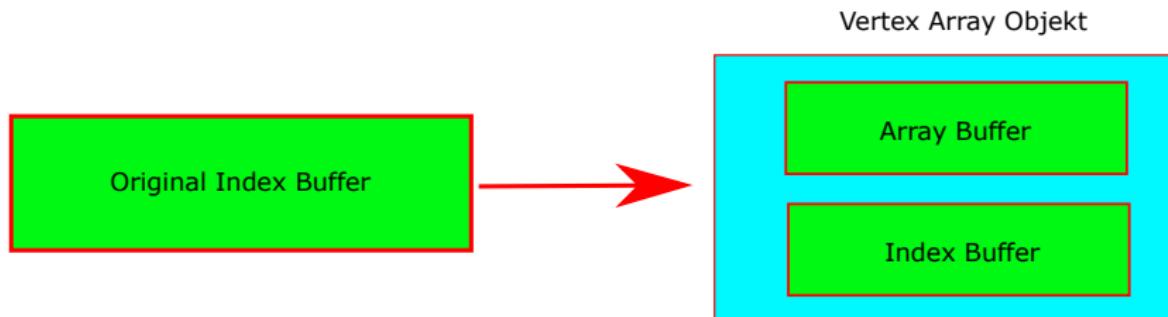
Ausblick  
○○○○○○○○

Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

23. Juni 2019



# Primitive Assembly



Motivation  
○○

Anwendung  
○

Geometry Stage  
●○○○○○○○

Rasterisierung  
○

Per Frag Ops  
○

Koordinatensysteme  
○

Compute Shader  
○

Ausblick  
○○○○○○○○

Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

23. Juni 2019



# Vertex Shader

Eckpunkt in Modelkoordinaten  
und Attribute



## Vertex Shader

Eckpunkt in Clip Space



Motivation  
○○

Anwendung  
○

Geometry Stage  
○●○○○○○○

Rasterisierung  
○

Per Frag Ops  
○

Koordinatensysteme  
○

Compute Shader  
○

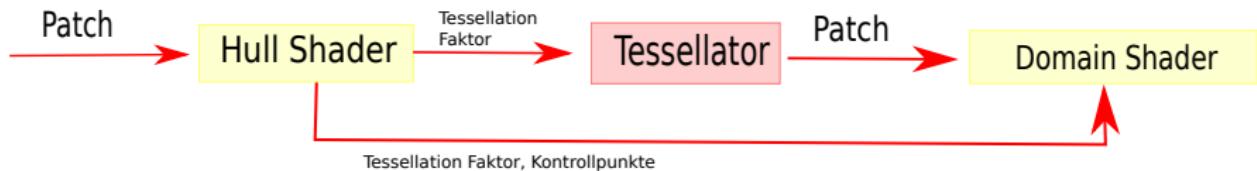
Ausblick  
○○○○○○○○○○

Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

23. Juni 2019



# Tessellation



Motivation  
○○

Anwendung  
○

Geometry Stage  
○○●○○○○○

Rasterisierung  
○

Per Frag Ops  
○

Koordinatensysteme  
○

Compute Shader  
○

Ausblick  
○○○○○○○○

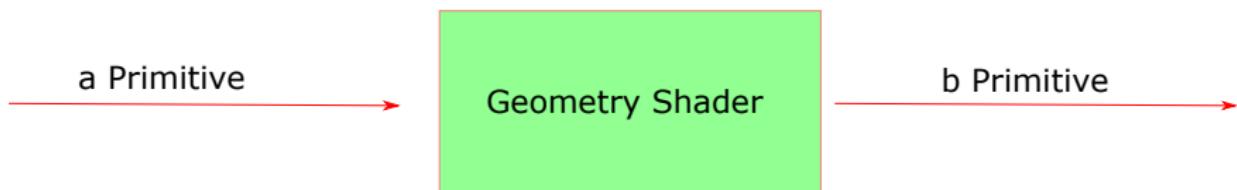
Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

23. Juni 2019

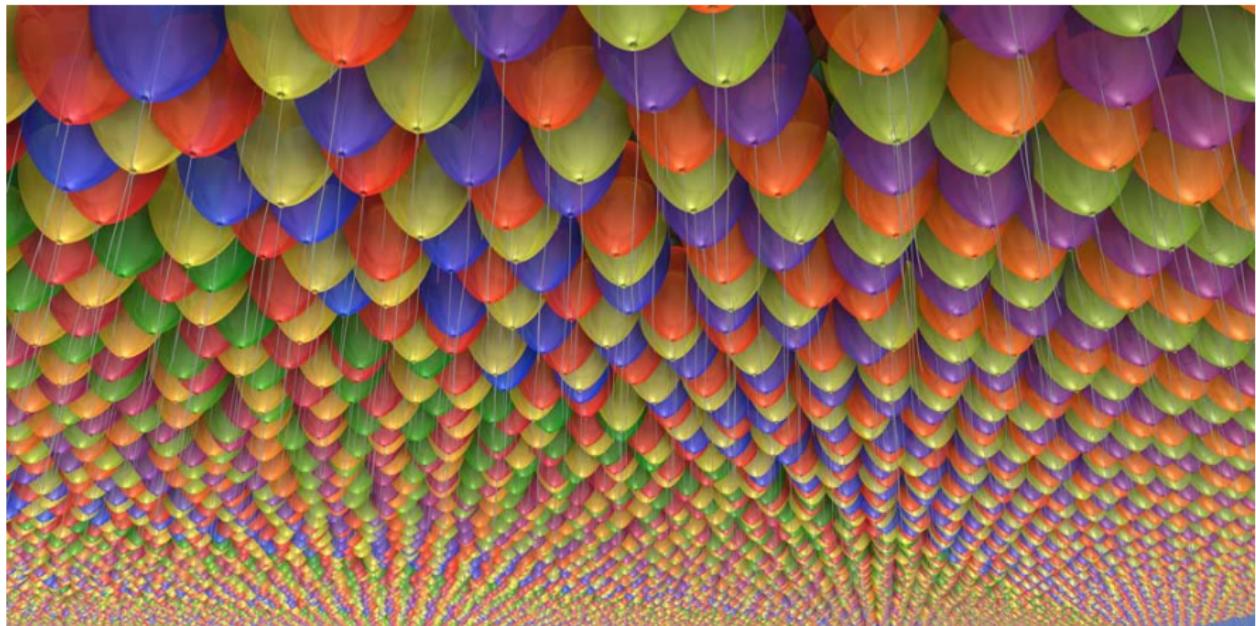


# Geometry Shader

Primitive (Punkt, Linie, Dreieck) werden vervielfacht, entfernt oder umgewandelt



# Geometry Shader - Instanziierung



25

Motivation

Anwendung

Geometry Stage

Rasterisierung

Per Frag Ops

Koordinatensysteme

Compute Shader

Ausblick

Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

23. Juni 2019

10/25



# Geometry Shader und Shading

```
layout (triangles) in;
layout (triangle_strip , max_vertices = 3 )out;
void main() {
    for(int j = 0; j < gl_in.length(); i++) {
        gl_Position = MVP * gl_in[j].gl_Position;
        vec3 P = vec3 ( MV * in_position );
        //Gourand Shading
        vec3 N = normalize ( vec3 ( matrixNormale
                            * vec4 ( in_normal , 0.0 ) ) )
        vec3 L = normalize ( lightSourcePos-P );
        color = vec4 ( max ( 0.0, dot( L, N ) )
        EmitVertex();
    }
    EndPrimitive();
}
```

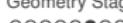
Motivation



Anwendung



Geometry Stage



Rasterisierung



Per Frag Ops



Koordinatensysteme



Compute Shader



Ausblick



Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

23. Juni 2019



# Projektionstransformation

Motivation	Anwendung	Geometry Stage	Rasterisierung	Per Frag Ops	Koordinatensysteme	Compute Shader	Ausblick
○○	○	○○○○○●○	○	○	○	○	○○○○○○○○

Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

23. Juni 2019

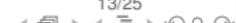


# Clipping

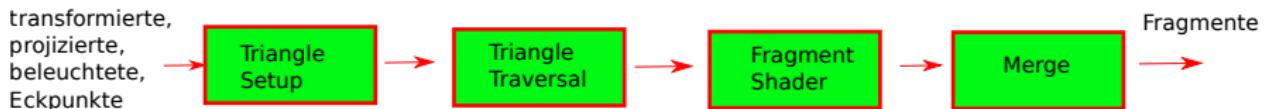
Motivation	Anwendung	Geometry Stage	Rasterisierung	Per Frag Ops	Koordinatensysteme	Compute Shader	Ausblick
○○	○	○○○○○○●	○	○	○	○	○○○○○○○○

Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

23. Juni 2019



# Rasterisierung



Motivation  
○○

Anwendung  
○

Geometry Stage  
○○○○○○○○

Rasterisierung  
●

Per Frag Ops  
○

Koordinatensysteme  
○

Compute Shader  
○

Ausblick  
○○○○○○○○

Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

23. Juni 2019



# Per Fragment Operations

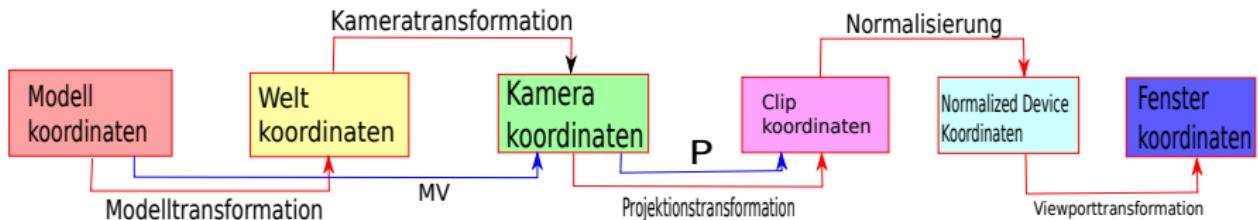
Motivation	Anwendung	Geometry Stage	Rasterisierung	Per Frag Ops	Koordinatensysteme	Compute Shader	Ausblick
○○	○	○○○○○○○○	○	●	○	○	○○○○○○○○

Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

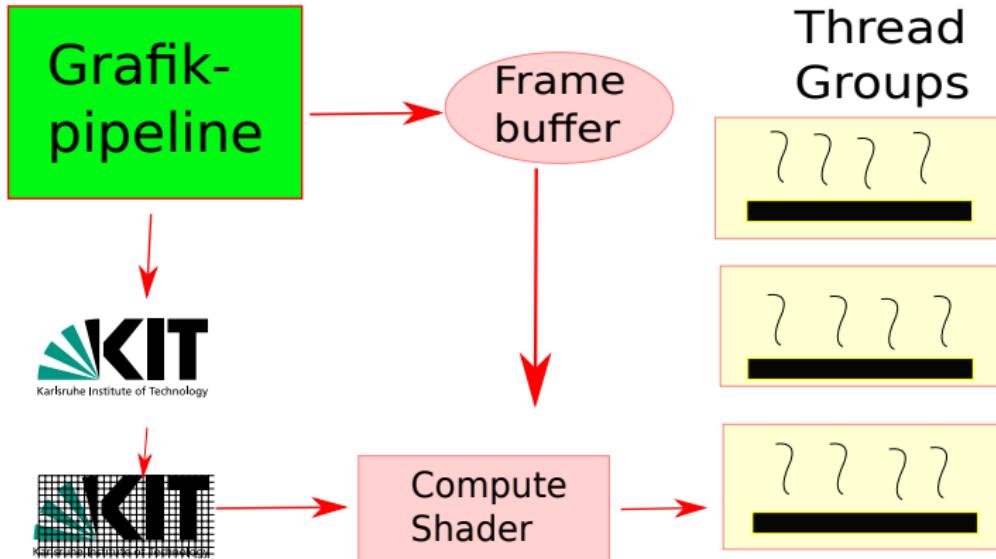
23. Juni 2019



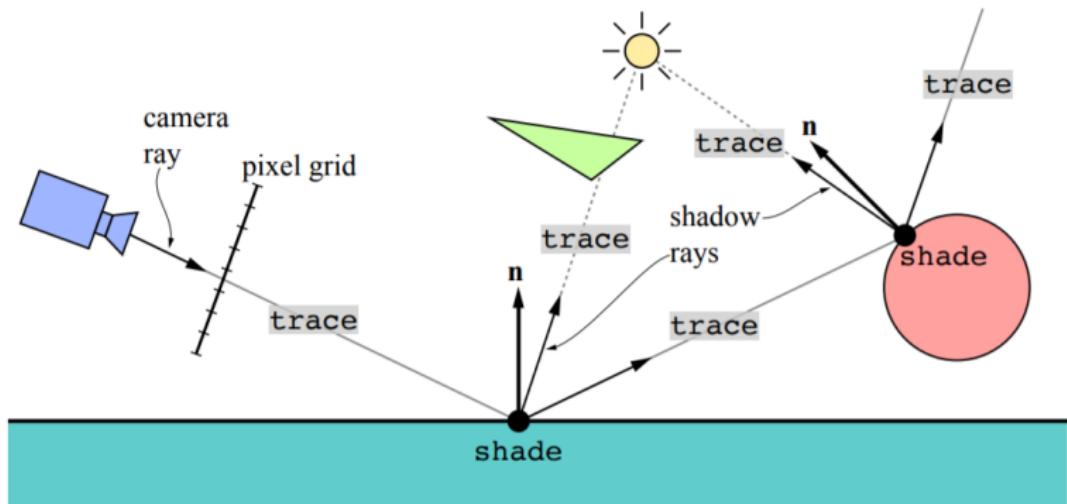
# Koordinatensysteme



# Compute Shader



## Raytracing Unterstützung



25

## Motivation

Anwendung

## Geometry Stage 88888888

## Rasterisierung

Per Frag Ops  
0

## Koordinatensysteme

## Compute Shader

## Ausblick

Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

23. Juni 2019

18/25

# Raytracing Unterstützung



25

Motivation

Anwendung

Geometry Stage

Rasterisierung

Per Frag Ops

Koordinatensysteme

Compute Shader

Ausblick

Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

23. Juni 2019

19/25



# Hybrides Rendering in *PICA PICA*

- Transparenz und Durchsichtigkeit Ray-Tracing
- Global Illumination Ray-Tracing Compute
- G-Buffer Layout Rasterisierung
- Direkte Schatten Ray-Tracing Rasterisierung
- Reflexionen Ray-Tracing Compute
- Direkte Beleuchtung Compute
- Post-Processing Compute

25

Motivation	Anwendung	Geometry Stage	Rasterisierung	Per Frag Ops	Koordinatensysteme	Compute Shader	Ausblick
○○	○	○○○○○○○○	○	○	○	○	○○●○○○○○○

Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

23. Juni 2019



# Hybrides Rendering in *PICA PICA*

- Transparenz und Durchsichtigkeit Ray-Tracing
- Global Illumination Ray-Tracing Compute
- G-Buffer Layout Rasterisierung
- Direkte Schatten Ray-Tracing Rasterisierung
- Reflexionen Ray-Tracing Compute
- Direkte Beleuchtung Compute
- Post-Processing Compute

25

Motivation	Anwendung	Geometry Stage	Rasterisierung	Per Frag Ops	Koordinatensysteme	Compute Shader	Ausblick
○○	○	○○○○○○○○	○	○	○	○	○○●○○○○○○

Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

23. Juni 2019



# Hybrides Rendering in *PICA PICA*

- Transparenz und Durchsichtigkeit Ray-Tracing
- Global Illumination Ray-Tracing Compute
- G-Buffer Layout Rasterisierung
- Direkte Schatten Ray-Tracing Rasterisierung
- Reflexionen Ray-Tracing Compute
- Direkte Beleuchtung Compute
- Post-Processing Compute

25

Motivation	Anwendung	Geometry Stage	Rasterisierung	Per Frag Ops	Koordinatensysteme	Compute Shader	Ausblick
○○	○	○○○○○○○○	○	○	○	○	○○●○○○○○○

Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

23. Juni 2019

20/25



# Hybrides Rendering in *PICA PICA*

- Transparenz und Durchsichtigkeit Ray-Tracing
- Global Illumination Ray-Tracing Compute
- G-Buffer Layout Rasterisierung
- Direkte Schatten Ray-Tracing Rasterisierung
- Reflexionen Ray-Tracing Compute
- Direkte Beleuchtung Compute
- Post-Processing Compute

25

Motivation	Anwendung	Geometry Stage	Rasterisierung	Per Frag Ops	Koordinatensysteme	Compute Shader	Ausblick
○○	○	○○○○○○○○	○	○	○	○	○○●○○○○○○

Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

23. Juni 2019



# Hybrides Rendering in *PICA PICA*

- Transparenz und Durchsichtigkeit Ray-Tracing
- Global Illumination Ray-Tracing Compute
- G-Buffer Layout Rasterisierung
- Direkte Schatten Ray-Tracing Rasterisierung
- Reflexionen Ray-Tracing Compute
- Direkte Beleuchtung Compute
- Post-Processing Compute

25

Motivation	Anwendung	Geometry Stage	Rasterisierung	Per Frag Ops	Koordinatensysteme	Compute Shader	Ausblick
○○	○	○○○○○○○○	○	○	○	○	○○●○○○○○○

Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

23. Juni 2019



# Hybrides Rendering in *PICA PICA*

- Transparenz und Durchsichtigkeit Ray-Tracing
- Global Illumination Ray-Tracing Compute
- G-Buffer Layout Rasterisierung
- Direkte Schatten Ray-Tracing Rasterisierung
- Reflexionen Ray-Tracing Compute
- Direkte Beleuchtung Compute
- Post-Processing Compute

25

Motivation	Anwendung	Geometry Stage	Rasterisierung	Per Frag Ops	Koordinatensysteme	Compute Shader	Ausblick
○○	○	○○○○○○○○	○	○	○	○	○○●○○○○○○

Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

23. Juni 2019



# Hybrides Rendering in *PICA PICA*

- Transparenz und Durchsichtigkeit Ray-Tracing
- Global Illumination Ray-Tracing Compute
- G-Buffer Layout Rasterisierung
- Direkte Schatten Ray-Tracing Rasterisierung
- Reflexionen Ray-Tracing Compute
- Direkte Beleuchtung Compute
- Post-Processing Compute

25

Motivation	Anwendung	Geometry Stage	Rasterisierung	Per Frag Ops	Koordinatensysteme	Compute Shader	Ausblick
○○	○	○○○○○○○○	○	○	○	○	○○●○○○○○○

Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

23. Juni 2019



# Task/-Mesh Shaders



**Level of detail: Dynamic**  
**Total asteroids: 198,679**  
**Drawn triangles: 35,908,694**  
**Max LOD triangles: 2,174,777,032,704**

25

## Motivation

## Anwendung

## Geometry Stage 00000000

## Rasterisierung

## Per Frag Ops

## Koordinatensysteme

## Compute Shader

## Ausblick

Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

23. Juni 2019

21/25

# Task-/Mesh Shaders

## MESHLETS

TRADITIONAL PIPELINE



TASK/MESH PIPELINE



Motivation  
○○

Anwendung  
○

Geometry Stage  
○○○○○○○○

Rasterisierung  
○

Per Frag Ops  
○

Koordinatensysteme  
○

Compute Shader  
○

Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

23. Juni 2019

22/25



# Fragen?



25

Motivation

Anwendung

Geometry Stage

Rasterisierung

Per Frag Ops

Koordinatensysteme

Compute Shader

Ausblick

Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

23. Juni 2019

23/25



## Weiterführende Literatur

- ① Quake2 Real-Time Raytracing Project Q2VKPT
- ② NVIDIA RTX Ray Tracing
- ③ RTX
- ④ Physically Based Rendering
- ⑤ Heise about NVIDIA Asteroid Demo
- ⑥ Link to very much information!
- ⑦ Vulkan Rendering Pipeline

Motivation  
○○

Anwendung  
○

Geometry Stage  
○○○○○○○○

Rasterisierung  
○

Per Frag Ops  
○

Koordinatensysteme  
○

Compute Shader  
○

Ausblick  
○○○○○○●○

Jonas Heinle – Aufbau einer modernen Rendering-Pipeline

23. Juni 2019

24/25



## Bilder

- Titelbild
  - Cyberpunk 2077
  - Kette
  - Pipeline
  - SEED's Project *PICA PICA*
  - NVIDIA Asteroid Scene
  - Witcher Endslide
  - Geometry Shader Instanziierung
  - Ray Tracing Prinzip

## Videos

- ## ① SEED's Project *PICA PICA*

Motivation & Leadership - Anwendung

## Nvidias Asteroids

Jonas Heipke - Aufbau einer modernen Rendering Pipeline

## Koordinatensysteme

6

## Compute Shader

6

Ausblick

## Ausblick

23. Juni 2019

