

WebGPU

Laurin Agostini | 60526
17.07.2020

Bachelorarbeit „WebGPU“
Kolloquium

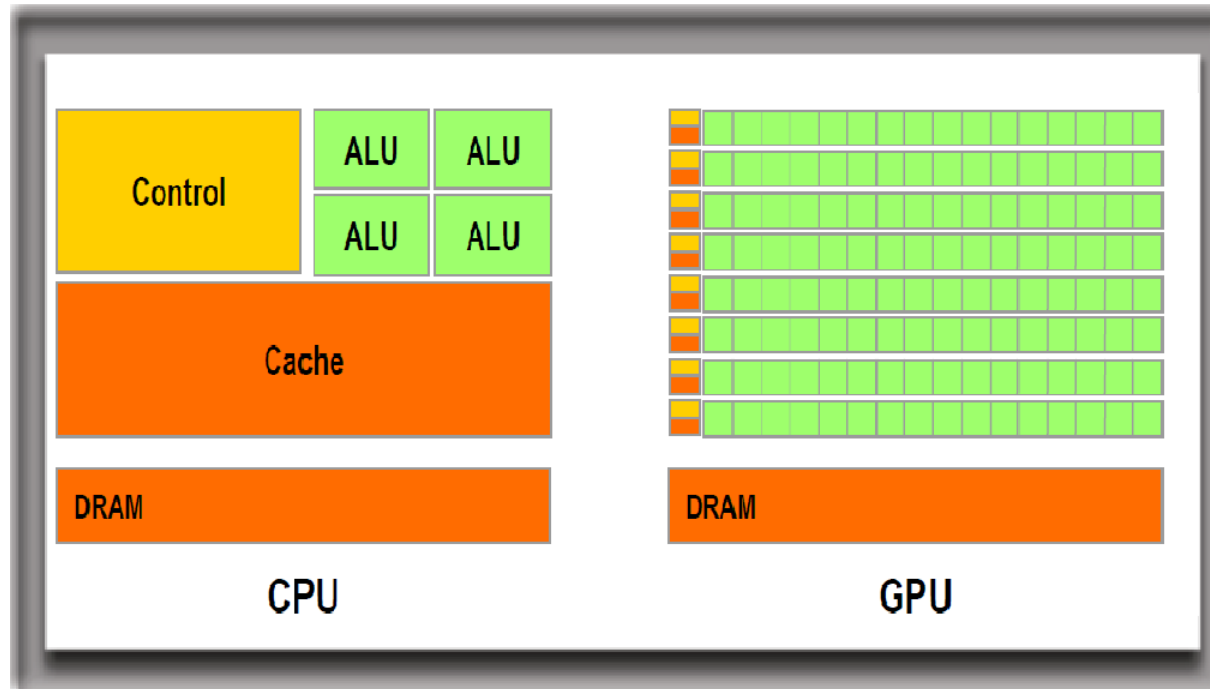
Gliederung

- Grundlagen
 - GPU
 - Rendering-Pipeline
 - Grafik-API
- WebGPU
 - Vergleich zu anderen Grafik-APIs
 - Aktueller Stand
 - Benutzung
- spider-Engine

GPU

- graphics processing unit
- Ausgelegt für parallele Prozesse und hohen Datendurchsatz
 - Nicht jede Applikation profitiert davon
- Grafikkarten haben oft eigenen Arbeitsspeicher (VRAM)

CPU / GPU



Quelle: <https://www.omnisci.com/technical-glossary/cpu-vs-gpu>

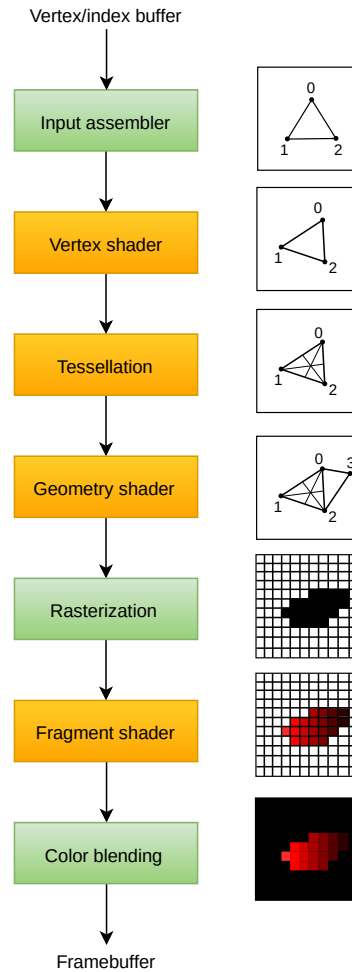
Grafikkarte



Quelle: https://commons.wikimedia.org/wiki/File:Zotac_Gaming_GTX_2080_ti.jpg

Rendering-Pipeline

- Beschreibt den Ablauf um aus Geometrie- und Texturdaten ein Bild zu erzeugen
- Früher fest in der Hardware verdrahtet
- Heutzutage nur noch wenige Teile fest

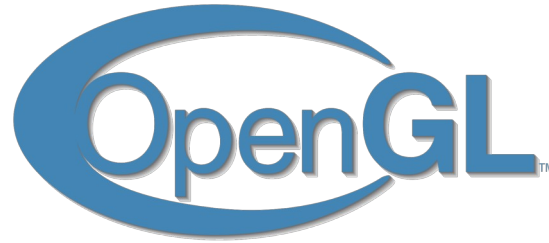


Quelle: https://vulkan-tutorial.com/images/vulkan_simplified_pipeline.svg

Grafik-API

- Programmierschnittstelle um (unter anderem) auf die Rendering-Pipeline einer GPU zuzugreifen
- Paradigmenwechsel in den letzten 6-7 Jahren
 - Mehr Kontrolle (aber auch mehr Aufwand) für Entwickler
 - Weniger Overhead im Treiber

Grafik-APIs



Microsoft®
DirectX®



Quellen:

<https://de.wikipedia.org/wiki/DirectX#/media/Datei:Microsoft-DirectX-Logo-wordmark.svg>

<https://de.wikipedia.org/wiki/OpenGL#/media/Datei:Opendgl-logo.svg>

https://de.wikipedia.org/wiki/WebGL#/media/Datei:WebGL_Logo.svg

„Moderne“ Grafik-APIs



Quellen:

<https://upload.wikimedia.org/wikipedia/commons/8/8b/Directx-12-logo-100251209-orig.jpg>

[https://de.wikipedia.org/wiki/Vulkan_\(API\)#/media/Datei:Vulkan.svg](https://de.wikipedia.org/wiki/Vulkan_(API)#/media/Datei:Vulkan.svg)

[https://en.wikipedia.org/wiki/Metal_\(API\)#/media/File:Apple_Metal_logo,_version_2.png](https://en.wikipedia.org/wiki/Metal_(API)#/media/File:Apple_Metal_logo,_version_2.png)

WebGPU

- Arbeitstitel für den Nachfolger von WebGL
- Erste Vorschläge und Anregungen 2016/2017
- Mitte 2018 Beginn erster Implementierungen

Vergleich zu anderen Grafik-APIs

- Inspiriert von den „modernen“ Grafik-APIs Vulkan, DirectX 12 und Metal
- Aber anders wie WebGL **keine** direkte Portierung einer bestehenden API (OpenGL ES)
- Im Vergleich zu den anderen APIs angepasst an das Web
 - WGSL statt Binär-Format für Shader
 - Mehr Fokus auf Sicherheit

Aktueller Stand

- Implementierungen in den Vorschauversionen der größten Webbrowser verfügbar
- Spezifikation ist noch nicht fertig und wird ständig angepasst und erweitert
- Experimentelle Unterstützung in wenigen 3D-JavaScript Frameworks (zum Beispiel `babylon.js`)

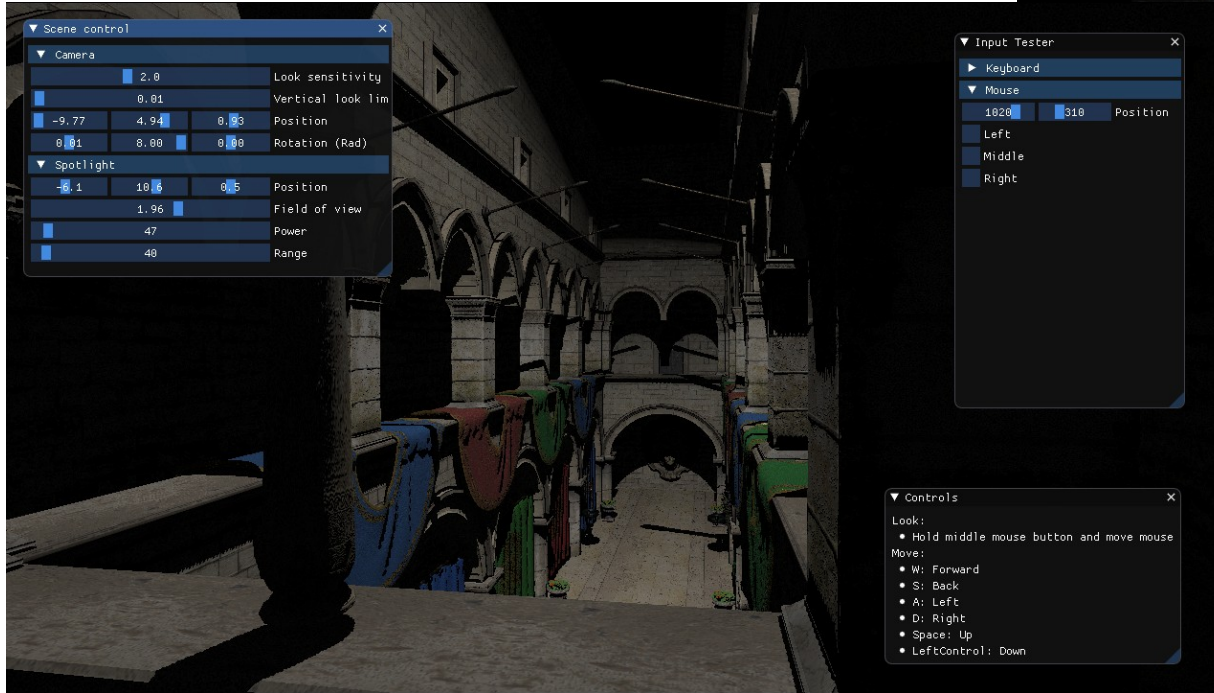
Benutzung

- Benötigt WebGPU fähigen Webbrowser
 - Chrome/Edge Canary
 - Firefox Nightly
 - (Safari Technology Preview)
- Einstiegspunkt `navigator.gpu`

Benutzung

- Ziemlich aufwendig zu initialisieren
 - 250-300 Zeilen JavaScript für ein Minimalbeispiel
 - Einerseits generell explizierter und andererseits wird versucht so viel Arbeit wie möglich zu Beginn zu erledigen
 - Ähnlich wie bei Vulkan und DirectX 12
- Handhabung von *Buffern* ist umständlich
 - Dies ist aber auch noch nicht fertig spezifiziert

spider-Engine



spider-Engine

- Knapp 4.000 Zeilen C99-Code
- Dazu kommen ~350 Zeilen GLSL-Shadercode
- Wird mithilfe von emscripten in HTML, JavaScript und WebAssembly umgewandelt



Quelle: https://emscripten.org/_static/Emscripten_logo_full.png

spider-Engine

Demo

Vielen Dank für Ihre Aufmerksamkeit!