|  |
| --- |
| Chromium Bibliothek  Analyse  **Julien Villiger**  **v1.1 01.03.2015** |
| **Berner Fachhochschule**  Technik und Informatik |

Inhaltsverzeichnis

[1 Architektur 3](#_Toc412989462)

[2 Anwendung 4](#_Toc412989463)

[2.1 Multi-Monitor Displays 4](#_Toc412989464)

[2.2 Delegation 4](#_Toc412989465)

[2.3 Manipulation 4](#_Toc412989466)

[2.4 Stereoskopie 4](#_Toc412989467)

[2.5 Command Stream Aufteilung 4](#_Toc412989468)

[3 Argumentation 5](#_Toc412989469)

[3.1 Pro 5](#_Toc412989470)

[3.1.1 Stereoskopie 5](#_Toc412989471)

[3.1.2 Infrastruktur 5](#_Toc412989472)

[3.1.3 Aufteilung der Monitore 5](#_Toc412989473)

[3.2 Kontra 5](#_Toc412989474)

[3.2.1 Plattform 5](#_Toc412989475)

[3.2.2 Maintenance 5](#_Toc412989476)

[3.2.3 OpenGL Support 5](#_Toc412989477)

[3.2.4 Kompatibilität Unity3D 5](#_Toc412989478)

[3.2.5 Netzwerkauslastung 5](#_Toc412989479)

[3.2.6 Community 5](#_Toc412989480)

[4 Zusammenfassung 6](#_Toc412989481)

[5 Literaturverzeichnis 7](#_Toc412989482)

[6 Versionskontrolle 7](#_Toc412989483)

# Architektur

Chromium ist eine OpenGL Implementation. Doch entgegen üblicher Implementationen, wird der OpenGL Command nicht in ein Rasterbild umgewandelt, sondern wird manipuliert und an andere OpenGL Implementationen weitergeschickt.

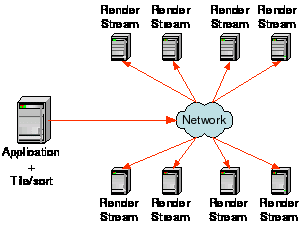
Die Chromium Bibliothek unterstützt eine Server / Client Architektur. Die Verarbeitungskette ist unterteilt in mehrere Stream Processing Units, kurz SPU.

Für jeden OpenGL Command hat eine SPU folgende Möglichkeiten:

* Modifizieren
* Ablehnen
* An eine weitere SPU weiterschicken

Die letzte SPU hat die Wahl, den OpenGL Command an eine lokale OpenGL Implementation zu überreichen um ein Rasterbild zu generieren, oder über ein Netzwerk an einen oder mehrere Chromium Servers zu schicken.

Die Chromium-Instanz läuft auf dem sogenannten „Mothership“ und managed die SPU Kette und Netzwerkverbindungen. Die laufende Applikation setzt die Drawcalls an die Hauptinstanz (Mothership) ab.

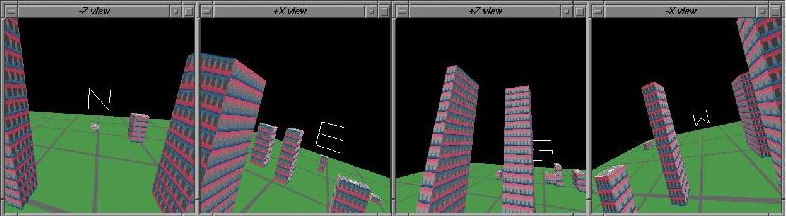


**Abbildung 1.0**, Architektur von Chromium mit Mothership und SUPs Quelle: http://chromium.sourceforge.net/doc/index.html

# Anwendung

## Multi-Monitor Displays

Darstellung der OpenGL Commands auf mehreren Displays. Konzipiert für einen CAVE mit mehreren Leinwänden.



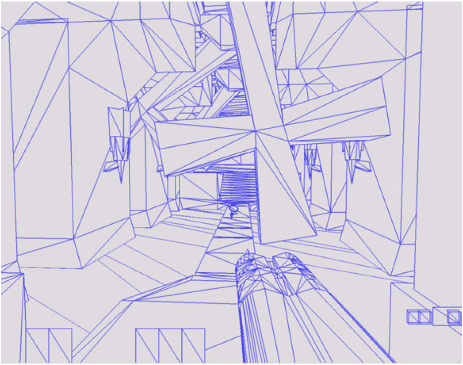
**Abbildung 1.1**, Ansicht der Leinwände in vier Himmelsrichtungen, Quelle: http://chromium.sourceforge.net/presentations/SantaFe-BrianPaul/siframes.html

## Delegation

Der gesamte OpenGL Stream kann von einem Rechner auf den anderen verschoben werden. Hat eine Maschine keine dedizierte Grafikkarte, kann diese Aufgabe an einen besser ausgerüsteten Rechner delegiert werden.

## Manipulation

Polygone eines OpenGL Streams können manipuliert werden. Sogar eigene Rendering Styles können dank der komplett programmierbaren Rendering Pipeline von Chromium implementiert werden.



**Abbildung 1.2**, Implementation eigener Rendering Styles, Quelle: (http://chromium.sourceforge.net/doc/index.html)

## Stereoskopie

Nicht-Stereoskopische Anwendungen können in Stereoskopische umgewandelt werden. Aktive (Shutter Glasses) sowie passive (Polarisierte Lichtprojektion) Stereoskopie werden unterstützt.

## Command Stream Aufteilung

Ein OpenGL Command Stream kann aufgeteilt werden, damit versch. Rechner einen Teil des Renderings übernehmen können. Vergleichbar mit nVidia’s SLI.

# Argumentation

## Pro

### Stereoskopie

Das wichtigste Feature, die Umwandlung in eine stereoskopische Darstellung, wird angeboten.

### Infrastruktur

Die benötigte Infrastruktur ist gegeben. Das Rendering kann auf mehrere Rechner verteilt werden.

### Aufteilung der Monitore

Jeder Projektor im CAVE ha einen eigenen Viewport und diese Aufteilung wird unterstützt.

## Kontra

### Plattform

Chromium wurde auf Linux entwickelt und sollte auf diesem System ausgeführt werden. Unter Windows und OSX gibt es bekannte Probleme. Das verlangte Linuxwissen müsste zusätzlich erarbeitet werden.

### Maintenance

Seit 2006 gab es kein Update mehr. In einer Präsentation aus dem Jahre 2004 wird als nächste Phase der Support von OpenGL 2.0 angestrebt (Quelle: Slide 27, <http://chromium.sourceforge.net/presentations/SantaFe-BrianPaul/siframes.html>). Bis anhin wurde dieser Task nicht umgesetzt.

Falls Anwender von Chromium auf ein Problem stossen, können Feature Requests auf sourceforge.net abgesetzt werden (<http://sourceforge.net/p/chromium/feature-requests/>). Die letzten Requests wurden im Jahre 2002 bearbeitet und geschlossen. Neu erstellte Einträge

Zitat auf der offiziellen Sourceforge-Seite (<http://sourceforge.net/projects/chromium/>), 22.01.2015:

„*UPDATE: Chromium is no longer updated or maintained. The project is frozen*.”

### OpenGL Support

Die letzte noch unterstützte OpenGL Version war 1.5 mit Chromium Release 1.5 (Dezember 2003). Die aktuelle Version von OpenGL ist 4.5 (Release August 2014).

### Kompatibilität Unity3D

Etliche Features, die über die OpenGL Version 1.5 hinausgehen und von Unity3D verwendet werden, könnten bei Chromium zu schwerwiegenden Problemen führen.

Der Output von Unity3D könnte inkompatibel mit den SUPs sein. Eine Modifikation des Outputs müsste in Betracht gezogen werden, wobei der Aufwand sehr schwer abschätzbar und nur bedingt zielführend ist.

### Netzwerkauslastung

Engpässe könnten entstehen, weil der gesamte OpenGL Stream übers Netz geschickt wird. Bei simplen Anwendungen mit wenigen Primitiven sollte die Performance ausreichen, in Anbetracht dessen, dass wir komplexe Unity3D Spiele rendern wollen, würde sicherlich die Netzwerkkapazität nicht ausreichen. Vorgängige Tests müssten durchgeführt werden.

### Community

Chromium hat keine aktive Community mehr, die bei Problemen bei der Installation Hilfestellung bieten könnte.

# Zusammenfassung

Basierend auf der Gegenüberstellung der Pro- und Kontra-Argumentation und obwohl unsere geforderten Key-Features von der Chromium Graphics Library abgedeckt werden, sind die Nachteile massiv überwiegend.

Ausschlaggebend sind in erster Linie der eingestellte Support und die fehlende Weiterentwicklung der Bibliothek. Solange Unity3D und OpenGL sich am Weiterentwickeln sind, müsste Chromium laufend nachziehen und die neu entwickelten Features unterstützen.

Das Ziel der Thesis ist der Einsatz moderner und zukunftsorientierter Technologien. Wird auf ein Relikt gesetzt, ist der Erfolg der Umsetzung fraglich und keinesfalls eine robuste Basis, um zeitgemässe Anwendungen laufen zu lassen.

# Literaturverzeichnis

**Chromium Website***(http://chromium.sourceforge.net/)*

**Chromium (computer graphics) - Wikipedia Eintrag** *(http://en.wikipedia.org/wiki/Chromium\_%28computer\_graphics%29)*

**Sourceforge**  
*(http://sourceforge.net/projects/chromium/)*

# Versionskontrolle

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Datum** | **Beschreibung** | **Autor** |
| 1.0 | 25.02.2015 | Dokument erstellt + erster Entwurf | Julien Villiger |
| 1.1 | 01.03.2015 | Dokument überarbeitet, erweitert und fertiggestellt | Julien Villiger |