

Interaktive Computergrafik



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



Interaktive Computergrafik

Übung 1

Prof. Dr. Frank Steinicke

Human-Computer Interaction, University of Hamburg



Interaktive Computergrafik

Übung 1

Organisatorisches

Zeitlicher Ablauf

*Hinweise zum Übungszettel,
ggf. theoretische Grundlagen*

*Vorstellung der
Lösungen*

VL (Di)

Ü (Do)

VL (Di)

Ü (Do)

*Herausgabe
Übungszettel*

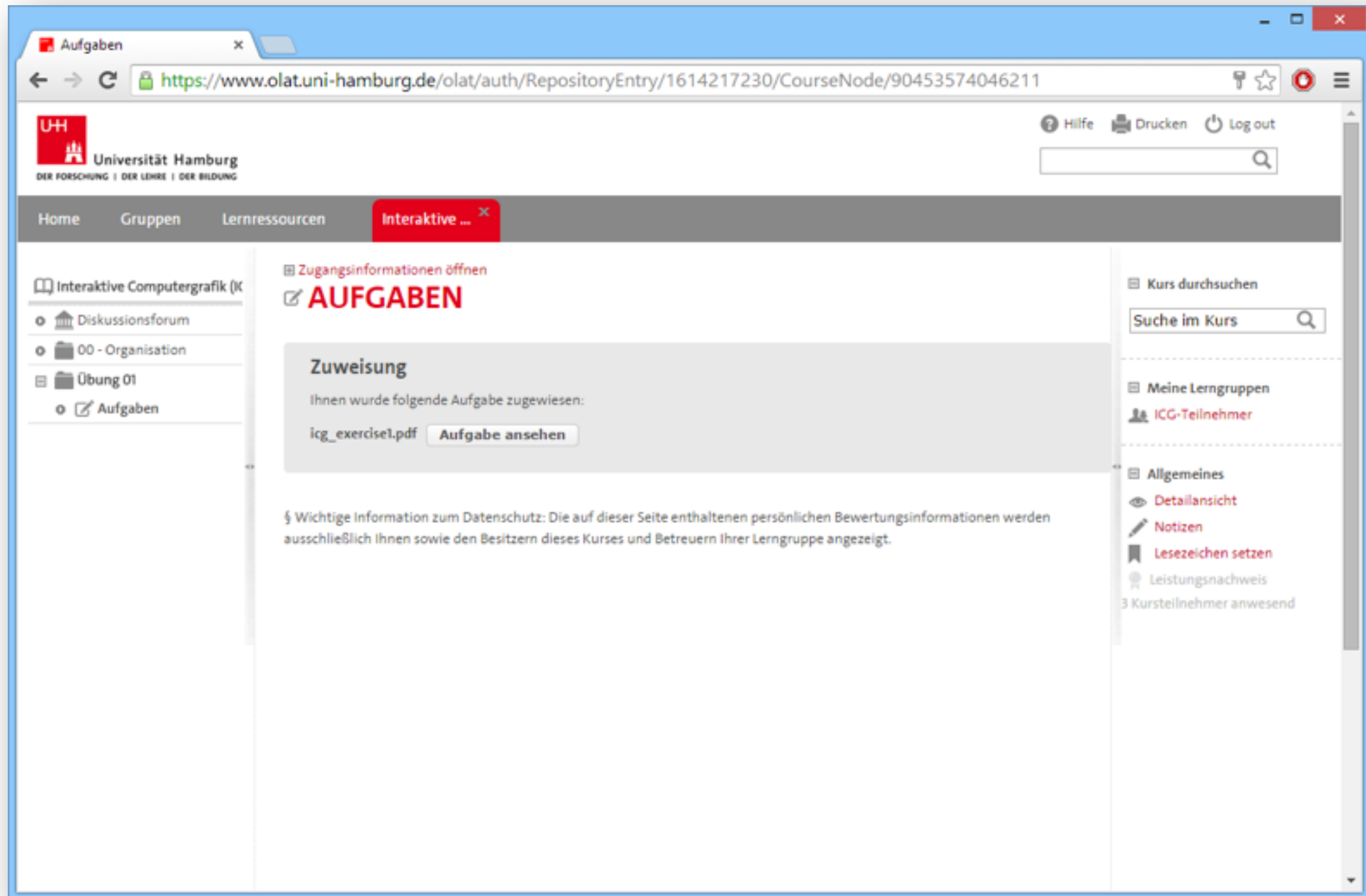
*Abgabe
Übungszettel*

OLAT

*Rückgabe kontrol-
lierter Lösungen*

*Veröffentlichung
Musterlösungen*

OLAT



Aufgabenbearbeitung



einzel

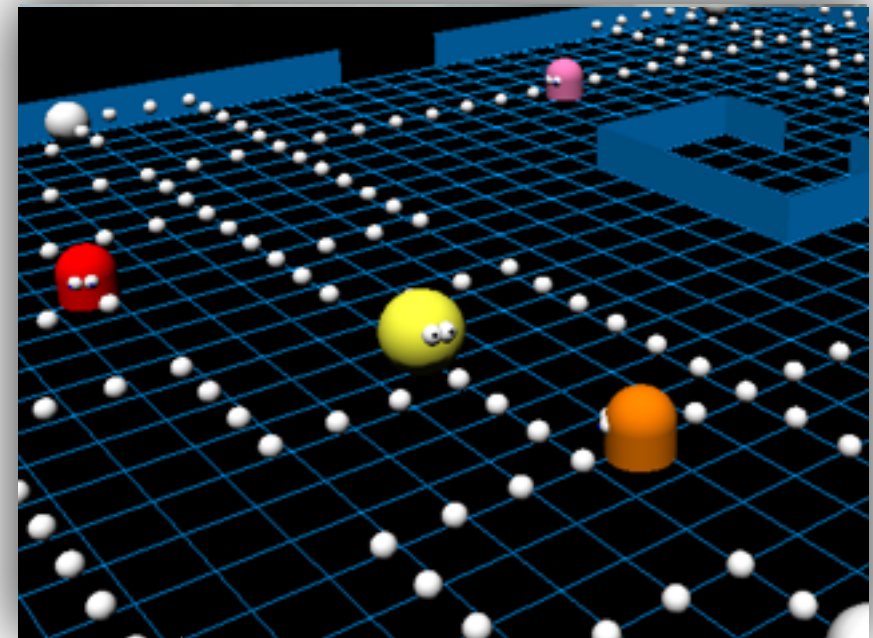


im Team

→ Bildung von festen **Gruppen** (je 3-4 Personen)

Übungsziel

Theoretische
Grundlagen aus der
Vorlesung



?





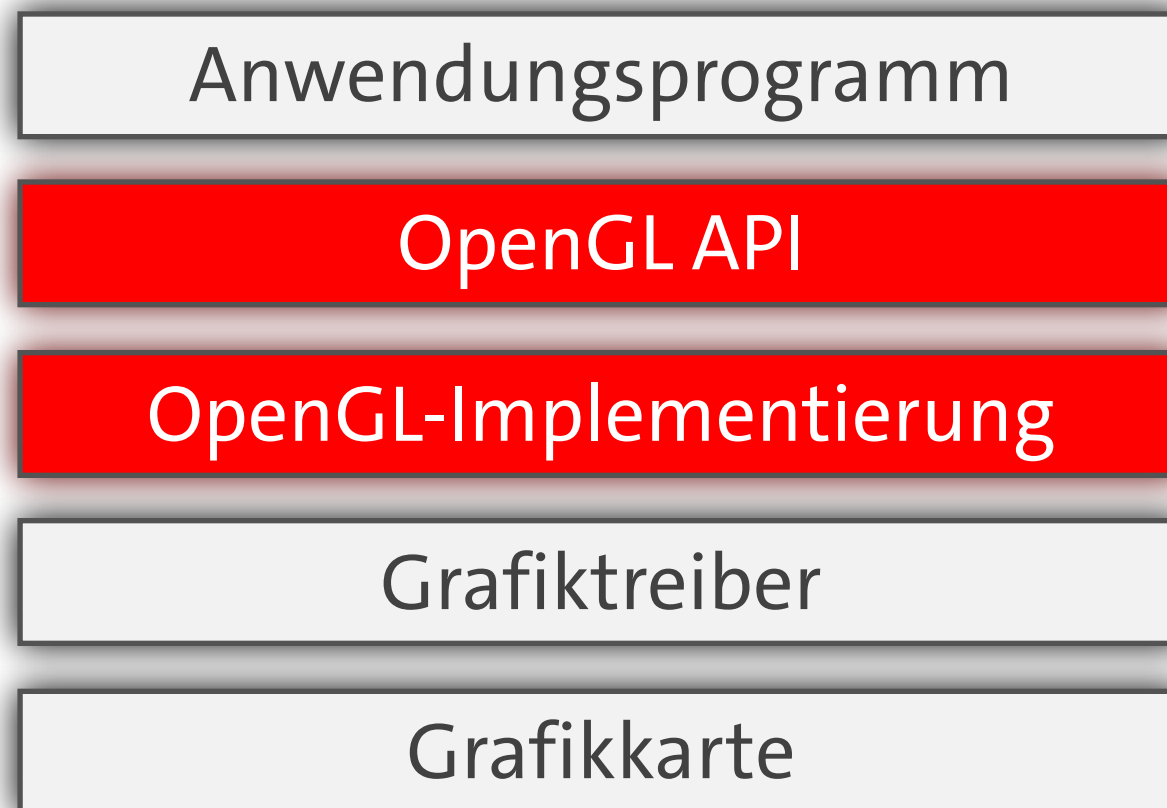
Interaktive Computergrafik

Übung 1

Grundlagen

OpenGL ist

- die **Spezifikation** einer plattform-unabhängigen Software-Schnittstelle für Grafik-Hardware



OpenGL ist nicht

- eine eigenständige Programmiersprache

→ “OpenGL-Programm” =

- C/C++-Programm, das OpenGL-Bibliothek(en) nutzt

+

- Shaderprogramme

OpenGL im Browser

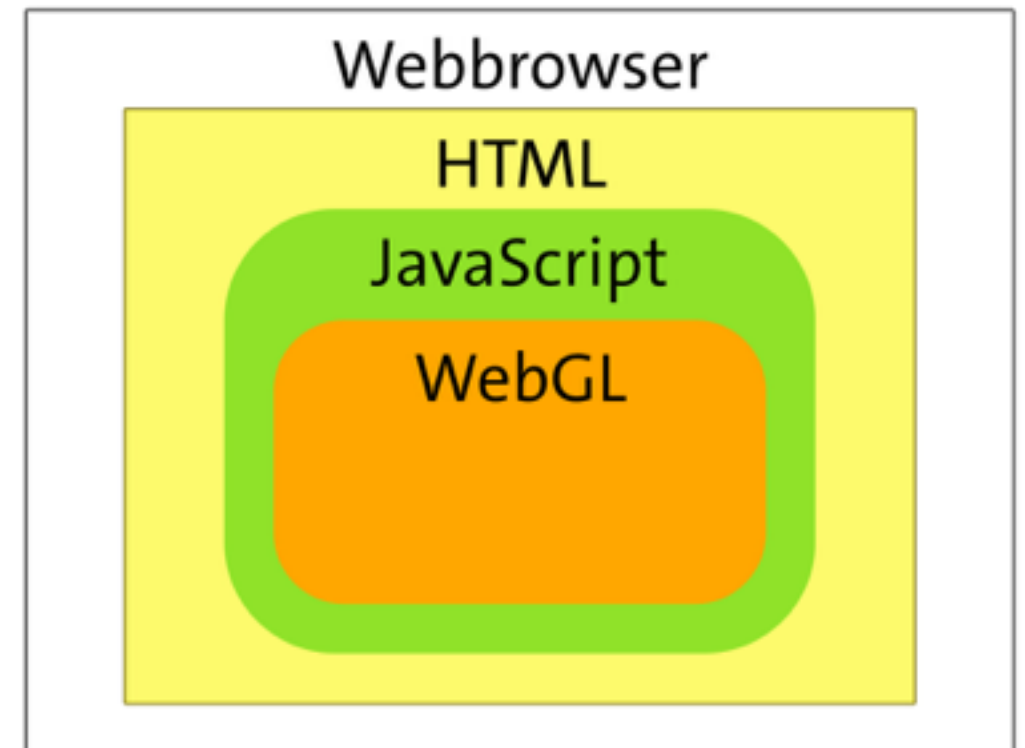
Geschichte

- Basis: **OpenGL ES** (Version 2.0)
 - reduzierte Version für eingebettete Systeme
- 2009: Gründung der „**WebGL Working Group**“ durch Khronos Group
 - Beteiligung von Mozilla, Apple, Google, AMD, Ericsson, Nvidia und Opera
- 2011: Release **WebGL 1.0**
 - Unterstützt von Chrome, Firefox, *Safari*, *Opera*

OpenGL im Browser

Prinzip

- Aufruf von JavaScript-Funktionen im HTML-Code
- Interpretation zur Laufzeit
- GPU Computing innerhalb von JavaScript Code → **hardwarebeschleunigte 3D-Grafiken im Browser**



OpenGL vs. WebGL

- WebGL unterstützt nicht alle OpenGL-Features, z.B.:
 - ▶ Double-precision floating-points
 - ▶ 3D Texturen
 - ▶ Geometry, Tessellation und Compute Shaders
 - ▶ Vertex Array Objects
 - ▶ Fixed Function Pipeline

Warum WebGL?

- Keine **Installation**
- Keine speziellen **Systemanforderungen**
- Keine zusätzlichen **Libraries** erforderlich
(z.B. für Behandlung von Nutzereingaben)
- **JavaScript** statt C++
- Gute Perspektiven für **Webentwickler**
- Gute Grundlage, um **OpenGL** zu lernen

Warum WebGL?

Browsersupport

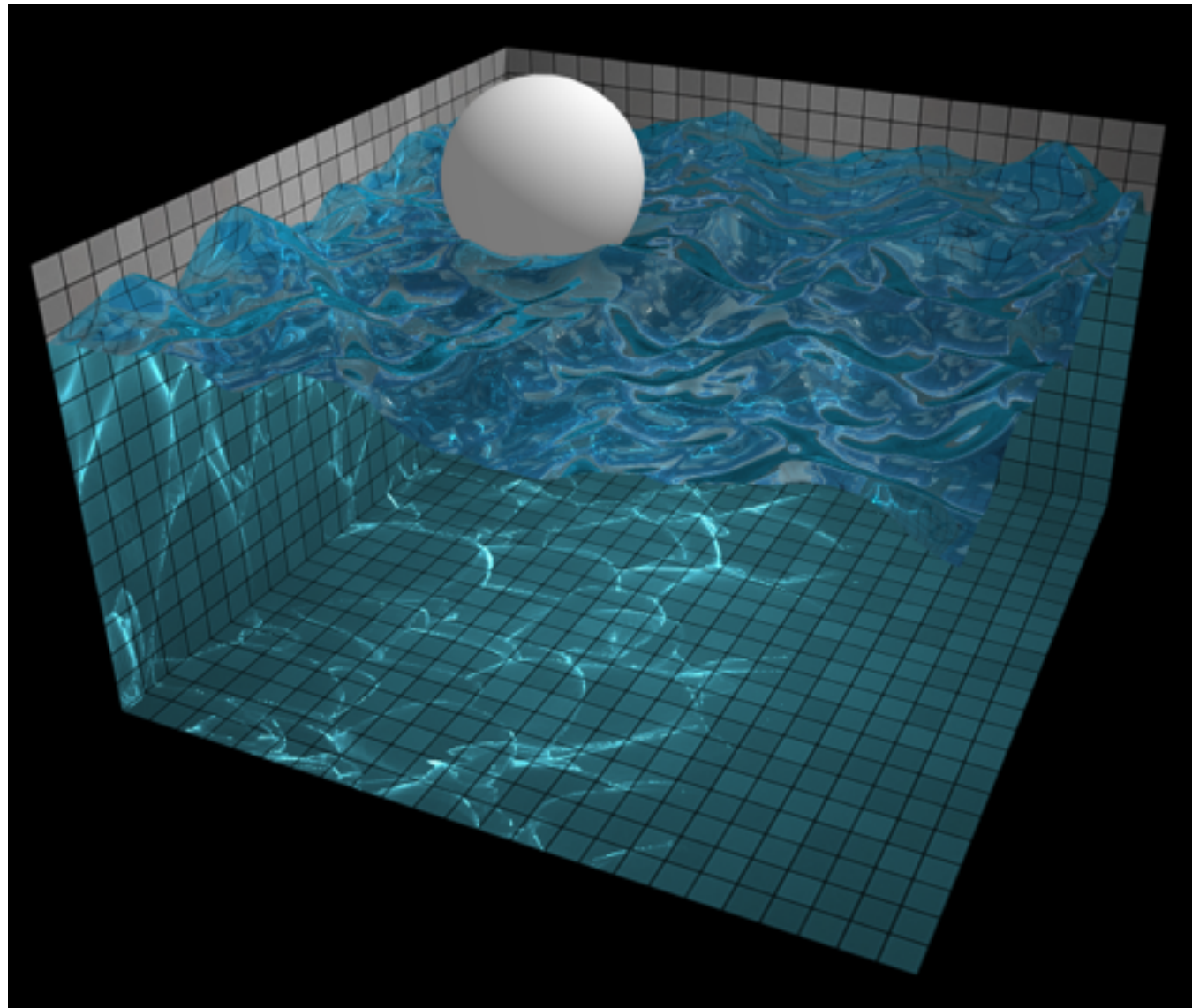
IE	Edge [*]	Firefox	Chrome	Safari	Opera	iOS Safari [*]	Opera Mini [*]	Android Browser [*]	Chrome for Android
			¹ 31					4.1	
8		38	43					4.3	
9		39	44					4.4	
10		40	45	8		8.4		4.4.4	
¹ 11	¹ 12	41	46	9	32	9	8	44	46
	¹ 13	42	47		33				
		43	48		34				
		44	49						

■ = Supported ■ = Not supported ■ = Partial support ■ = Support unknown

→ Hinweis: Korrektur/Test der Abgaben
immer in aktueller Version von **Chrome**!

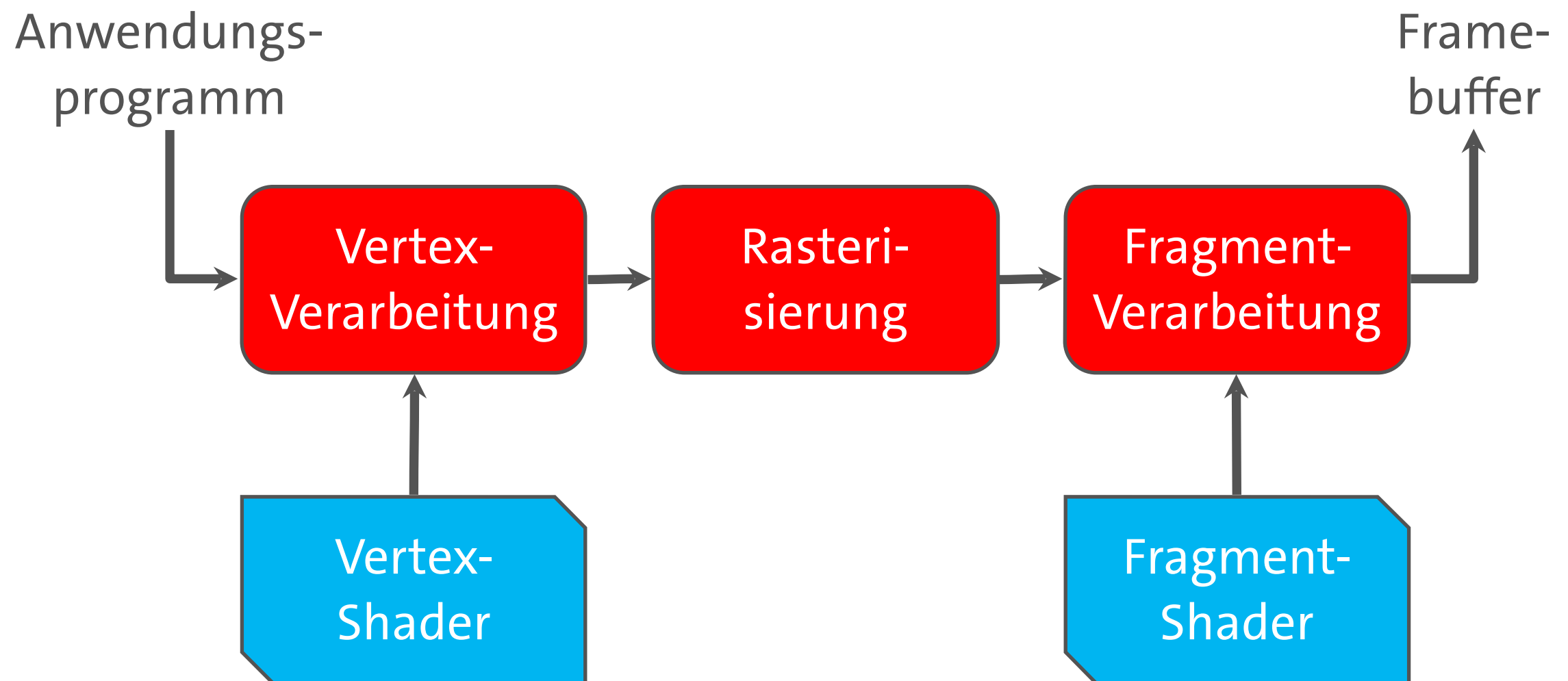
Warum WebGL?

State of the Art



Was behandeln wir?

WebGL-Pipeline (vereinfachte Version)



Was brauchen wir dafür?

Anwendungs-
programm

HTML + JavaScript (+ WebGL)

heute

WebGL API

nächste Woche

Vertex-
Shader

Fragment-
Shader

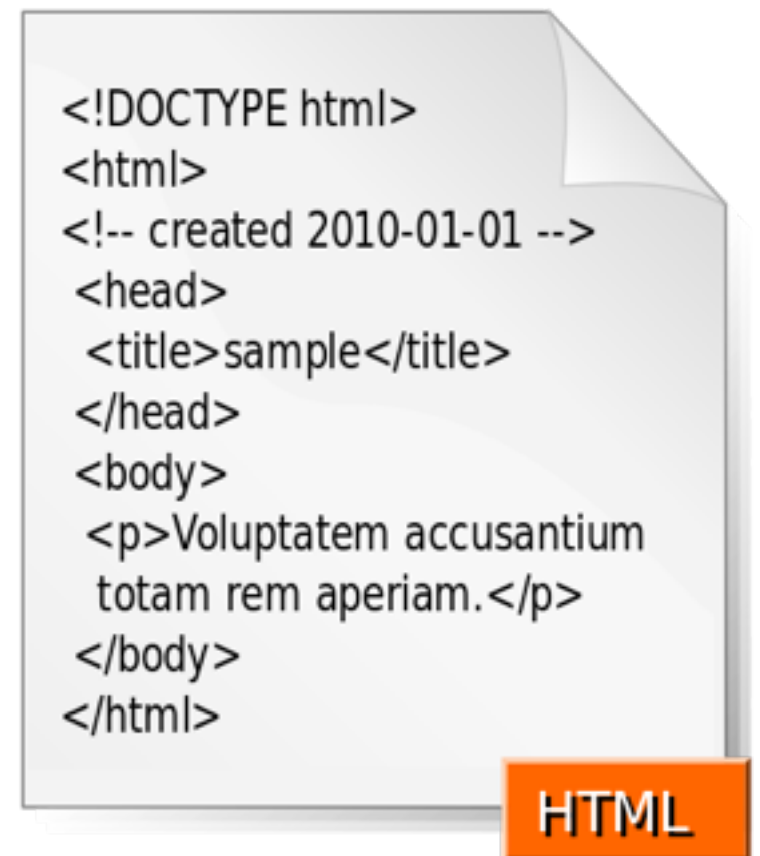
GLSL
(OpenGL Shading Language)

übernächste Woche



HTML

- Hypertext Markup Language
- Entstehungsidee:
 - Austausch von Forschungsergebnissen zwischen Mitarbeitern am CERN
- W3C / WHATWG
- Tag-Paare (Starttag/Endtag)



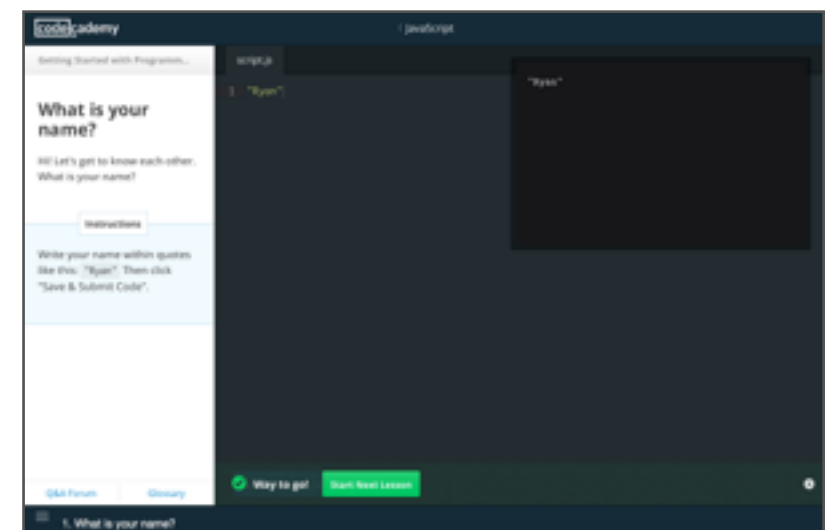
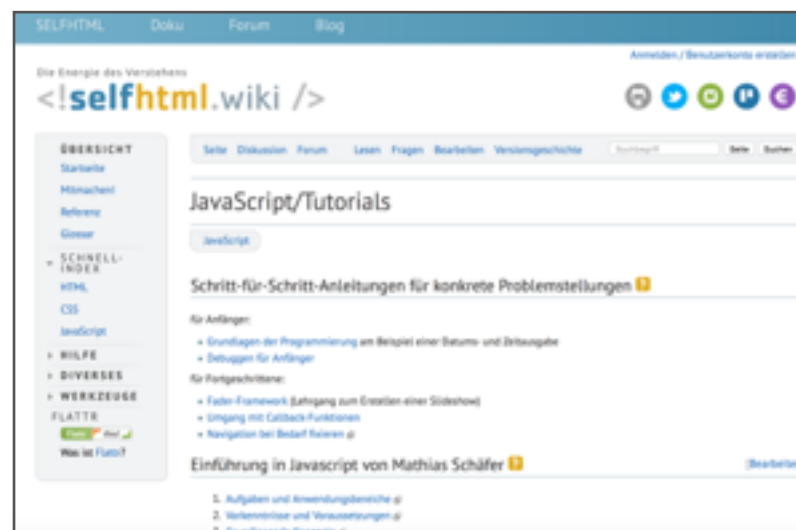
JavaScript

JS

- ursprünglich: Skriptsprache für dynamisches HTML in Webbrowsern
- **dynamisch typisierte**, objektorientierte, aber klassenlose Skriptsprache
- Programmierung **objektorientiert** und sowohl **prozedural** als auch **funktional**

JavaScript Tutorials

JS



- <http://www.w3schools.com/js/default.asp>
- <http://wiki.selfhtml.org/wiki/JavaScript/Tutorials>
- <http://www.codecademy.com/tracks/javascript>

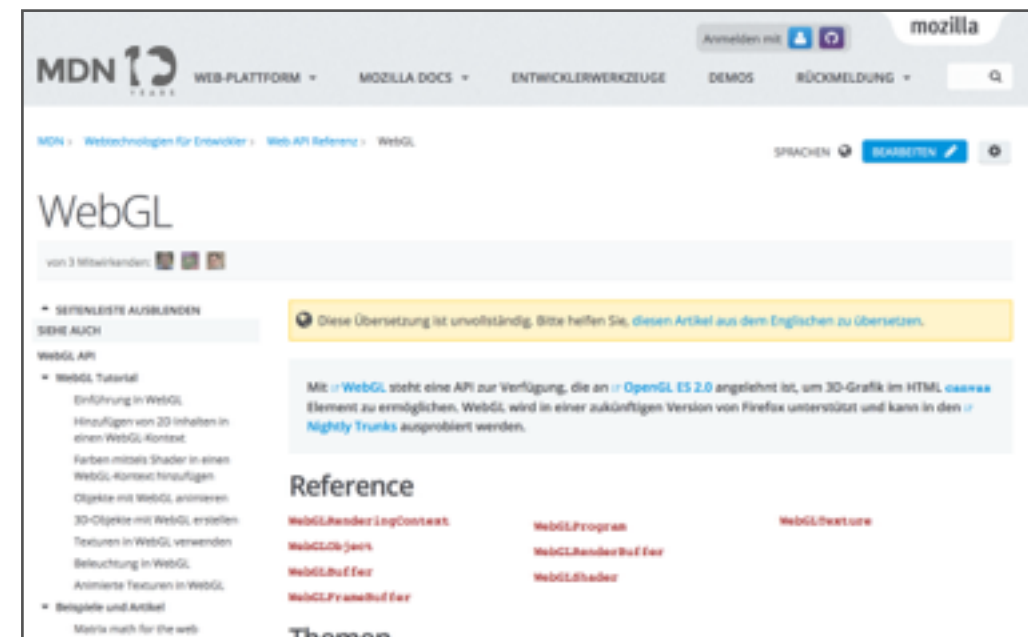
WebGL

Nützliche Links



- Browser Support Test: <http://get.webgl.org/>
- Debugging:
 - ▶ Allgemeine Informationen:
<https://www.khronos.org/webgl/wiki/Debugging>
 - ▶ Tool für Chrome:
<https://chrome.google.com/webstore/detail/webgl-inspector/ogkcjmbhnmfmlnielkjhedpcjomeaghda>
 - ▶ Tool für Firefox:
<https://addons.mozilla.org/en-US/firefox/addon/webgl-inspector/>

WebGL Tutorials



- <https://www.youtube.com/watch?v=tgVLb6fOVVc>
- http://developer.mozilla.org/de/docs/Web/API/WebGL_API

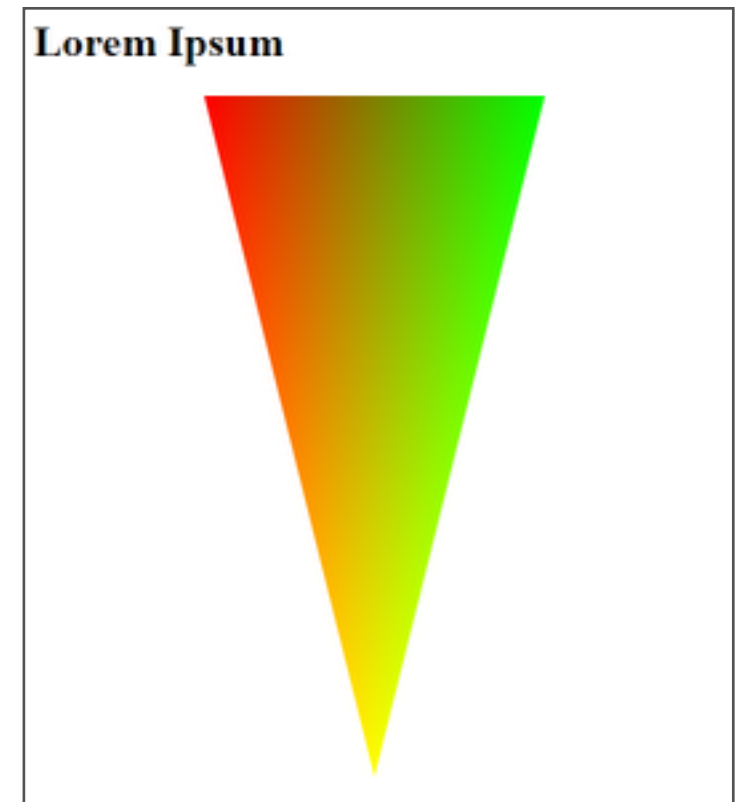
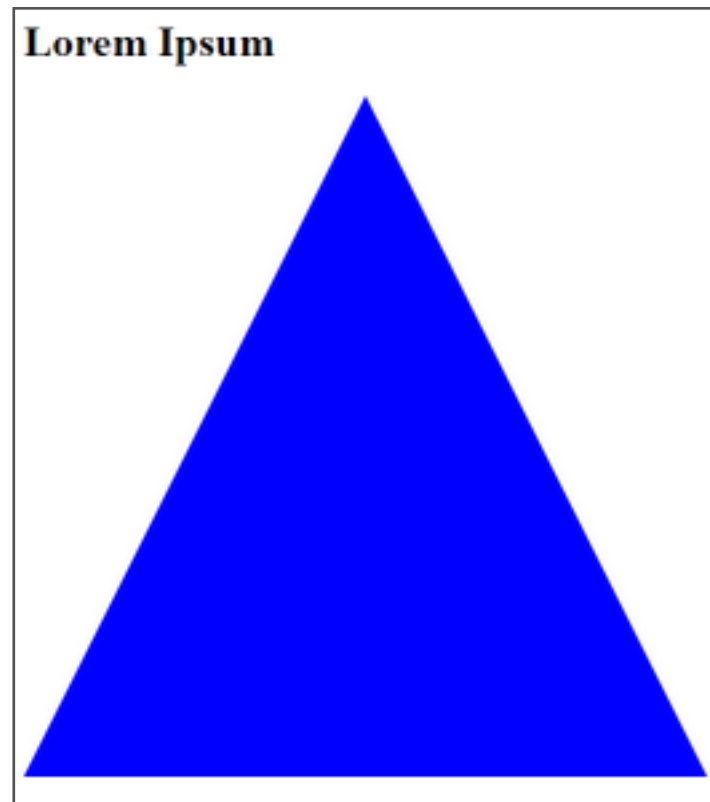
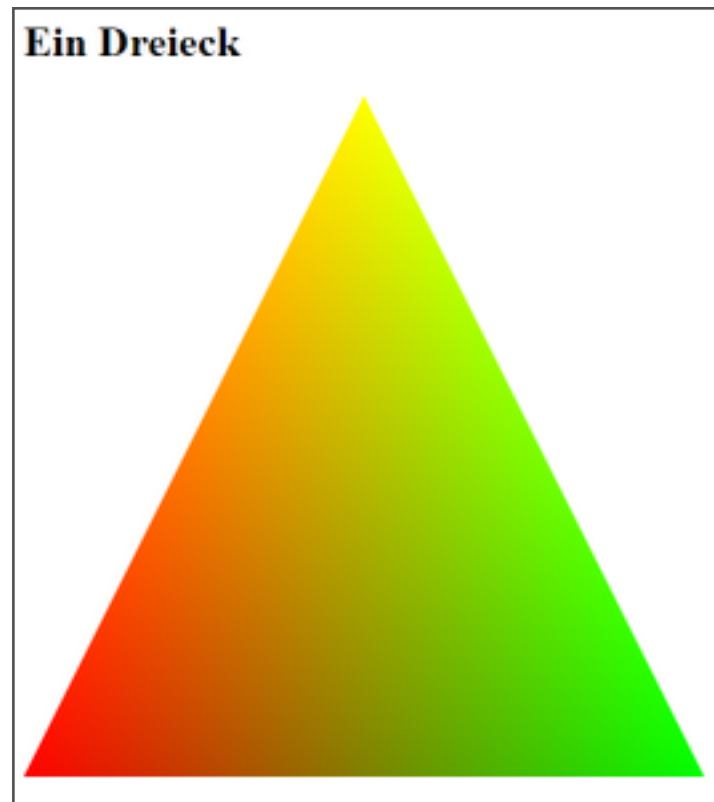


Interaktive Computergrafik

Übung 1

Übungszettel

triangles.js

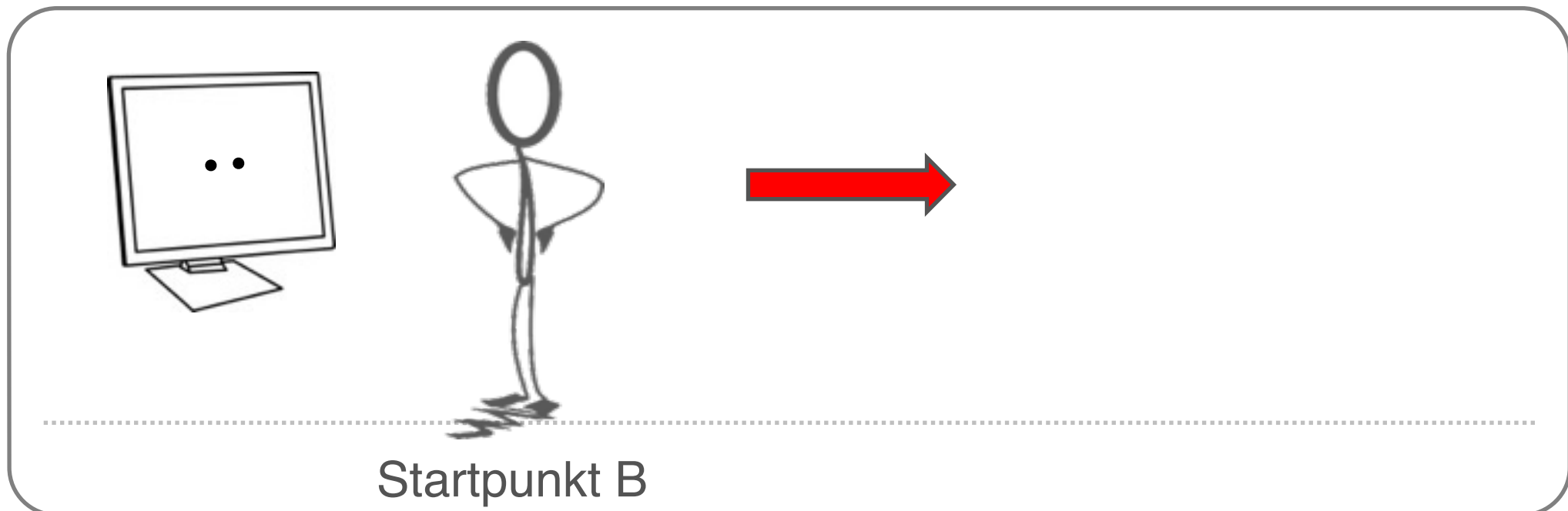
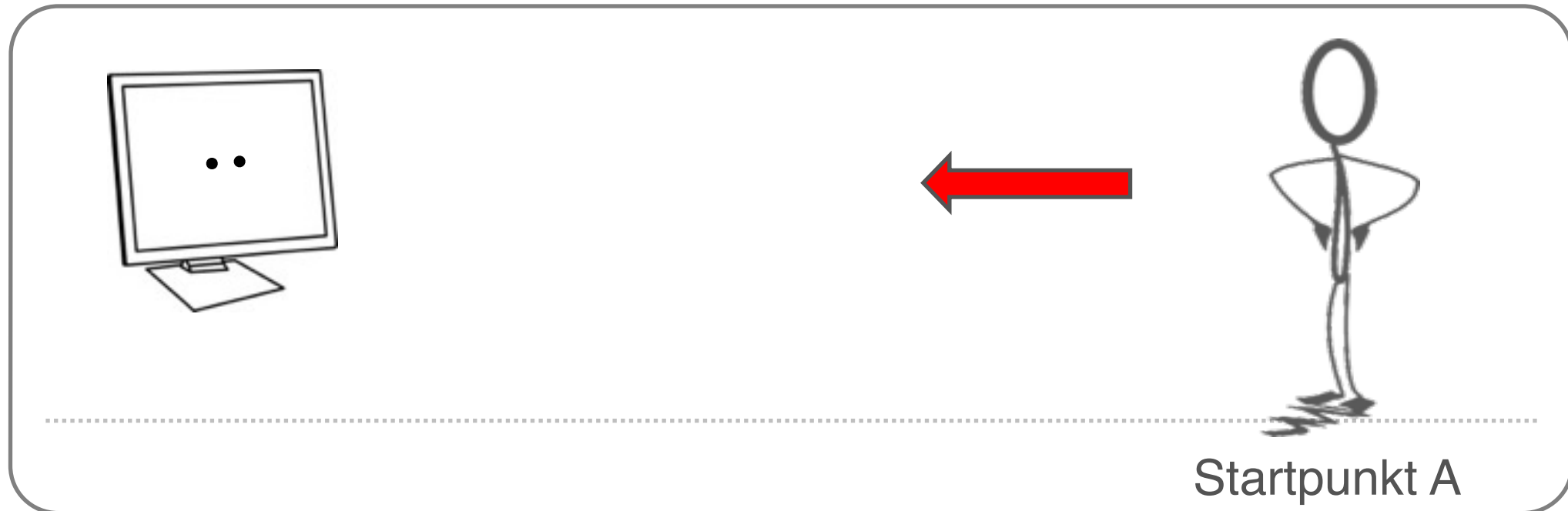


points.js

- 2 Punkte rendern
 - Durchmesser: 1 mm
 - Abstand: 2 mm

Hinweis: Abstand muss **per Hand** in Pixel umgerechnet werden, da Größe des Monitors dem Programm nicht bekannt ist

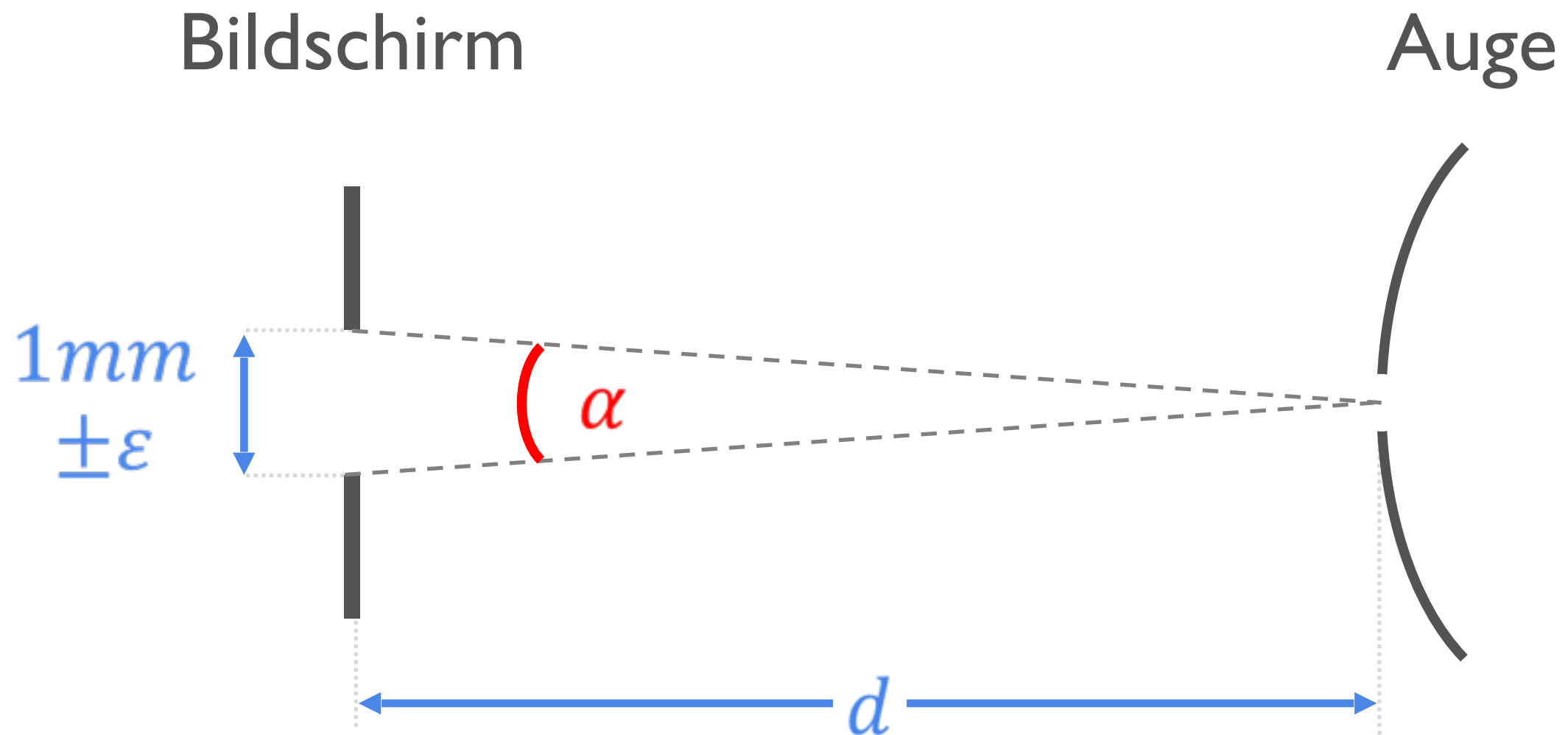
Experiment



Experiment



Experiment



Hinweis #1

- erst durchschnittliche Distanz pro Teilnehmer berechnen, dann über alle Teilnehmer mitteln

$$M = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^{10} d_j \quad n = \text{Anzahl der Teilnehmer}$$

Hinweis #2

- Standardabweichung der Stichprobe berechnen

$$S = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (M_i - M)^2}$$

n = Anzahl der Teilnehmer