WebGL Programming

Informatik
Prof. Dr. Thomas Koller
Dozent

T direkt +41 41 349 35 38
thomas.koller@hslu.ch

Joachim Wirth
Dozent

joachim.wirth@hslu.ch

OpenGL Geschichte

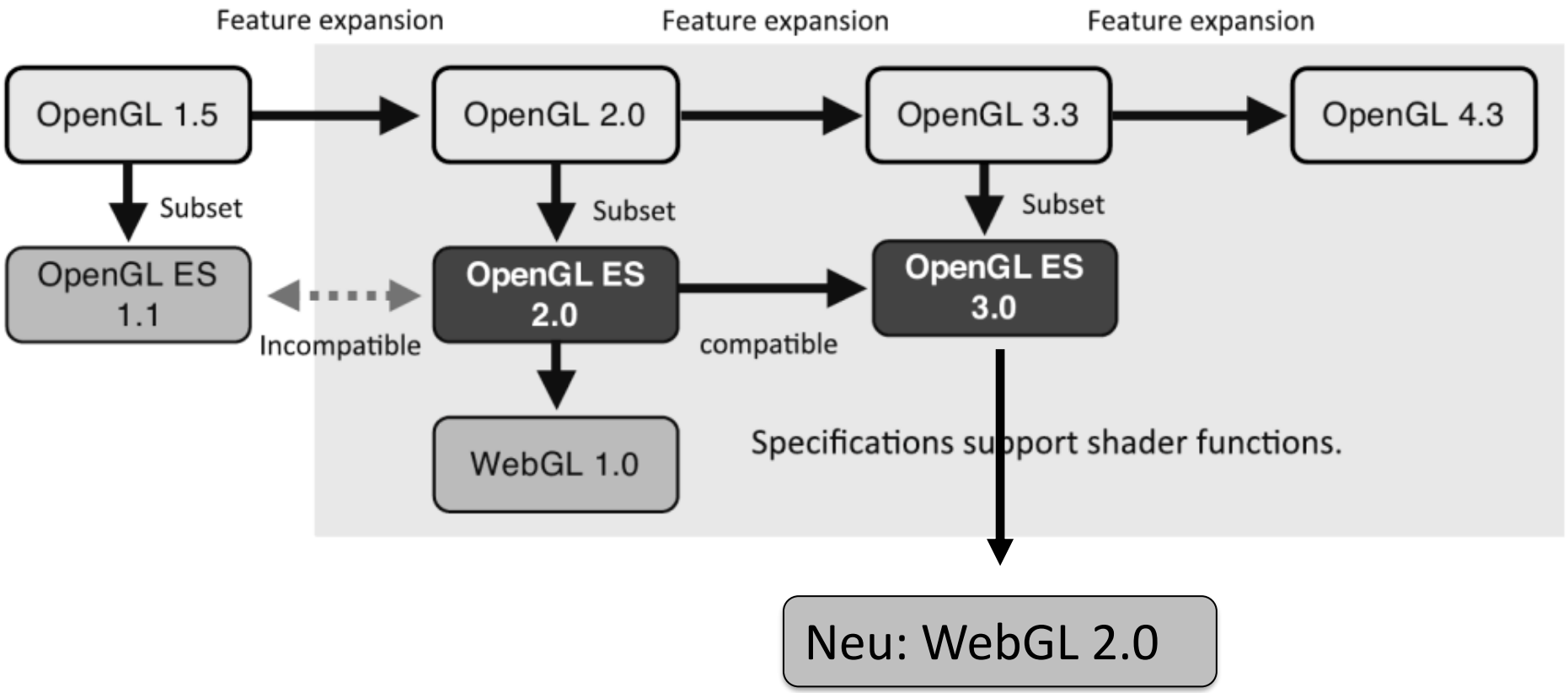
- 1992 aus IRIS GL (von SGI) entstanden
- verbreiteter als Standard Grafiksprache PHIGS
- Software und Hardware Unterstützung
- 1995: Microsoft Direct3D
- 1997 Joint Projekt „Fahrenheit“ von SGI und Microsoft (!)
- **OpenGL 2.0, 2004, Shader Language Support**
- OpenGL 3.0, 2008, zu wenig revolutionär
- **OpenGL 3.1, 2009, removed fixed function pipeline**
- OpenGL 4.6, 2017
- WebGL 2.0, 2017
- (Vulkan 1.0, 2016)

OpenGL Merkmale

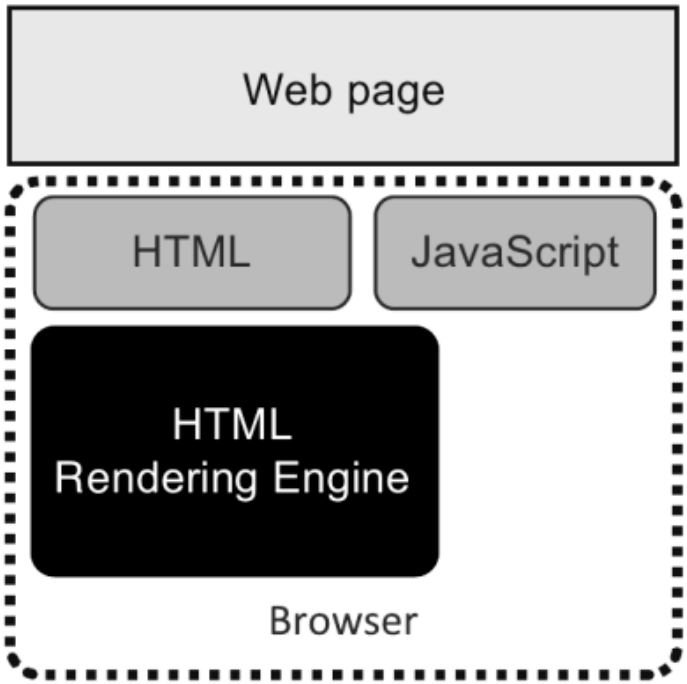
- Low Level Graphics API
- Unterstützung für verschiedene Plattformen
- gute Hardware-Unterstützung auf allen Grafikkarten
- Definierte Extensions
- Programmierbare Shader seit OpenGL 2.0
- Fixed Function Pipeline deprecated in OpenGL 3.0, gelöscht in OpenGL 3.1

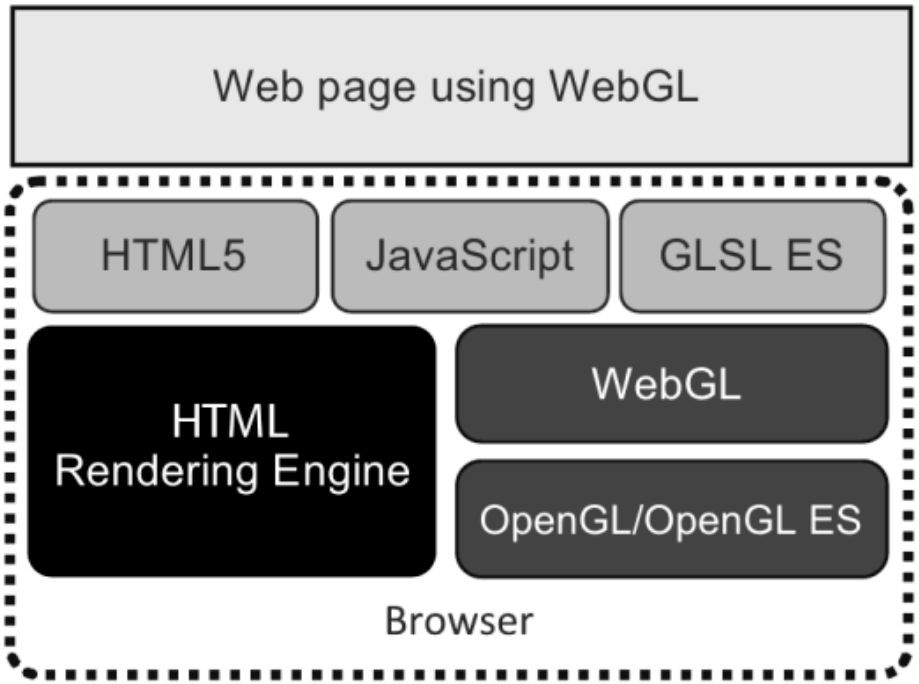
- Subset für Embedded Systems

Entwicklung von OpenGL



Einbetten von WebGL in Web Pages



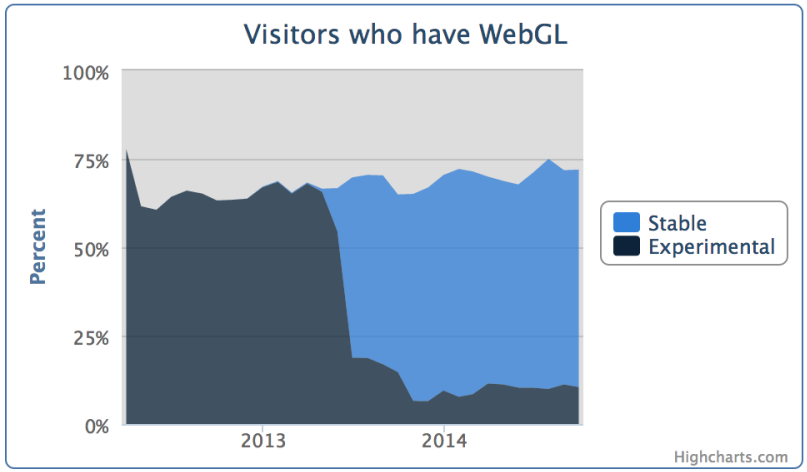


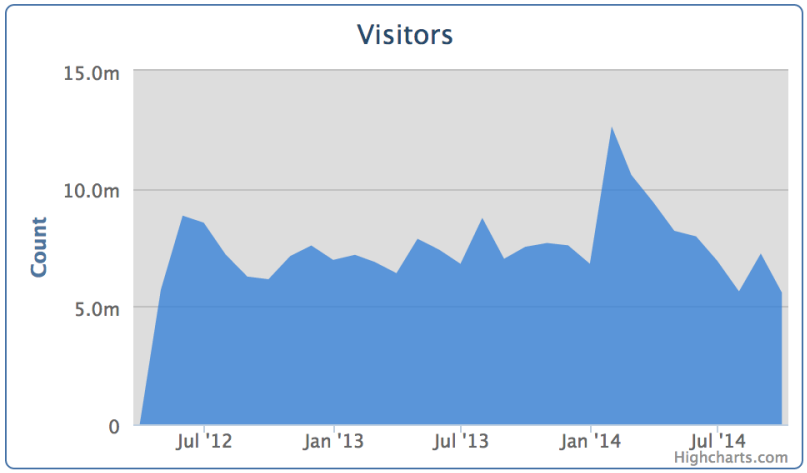
2014

- Personal computer 90%
  - Windows 81%
  - OS X 13%
  - Chrome OS 4%
  - Linux 2%
  - Other 0.4%
- Smartphone 6%
  - Android 55%
  - iOS 41%
  - Windows 2%
  - Other 2%
  - Firefox OS 0.0%
- Tablet 4%
  - iOS 88%
  - Android 11%
  - Windows 1.0%
  - Other 0.3%
- Game console 0.1%
  - Playstation 45%
  - Nintendo 35%
  - XBox 21%
  - Other 0.0%
- Smart TV 0.0%
  - Linux 99%
  - Other 0.6%
  - Windows 0.0%
- PDA 0.0%
  - Palm OS NaN%

WebGL Stats

Updated: Wednesday, 24 Sep 2014, 05:24AM UTC



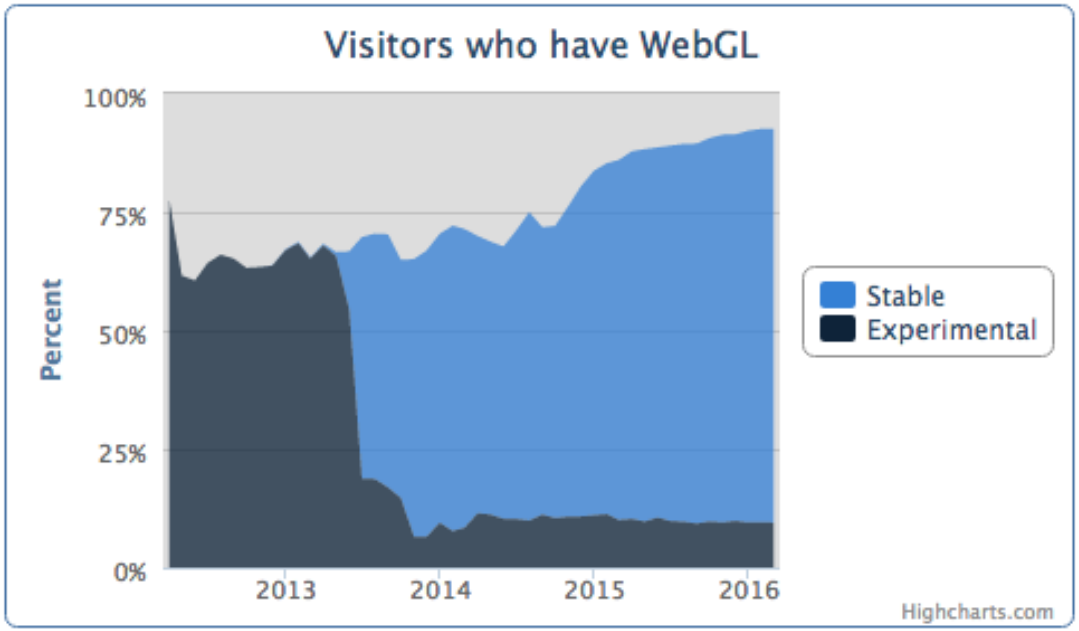


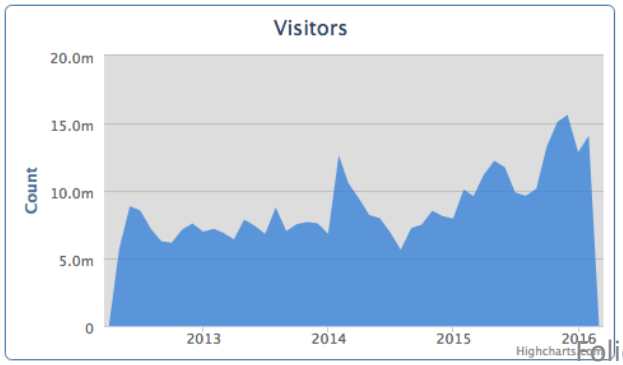
2016

WebGL Stats

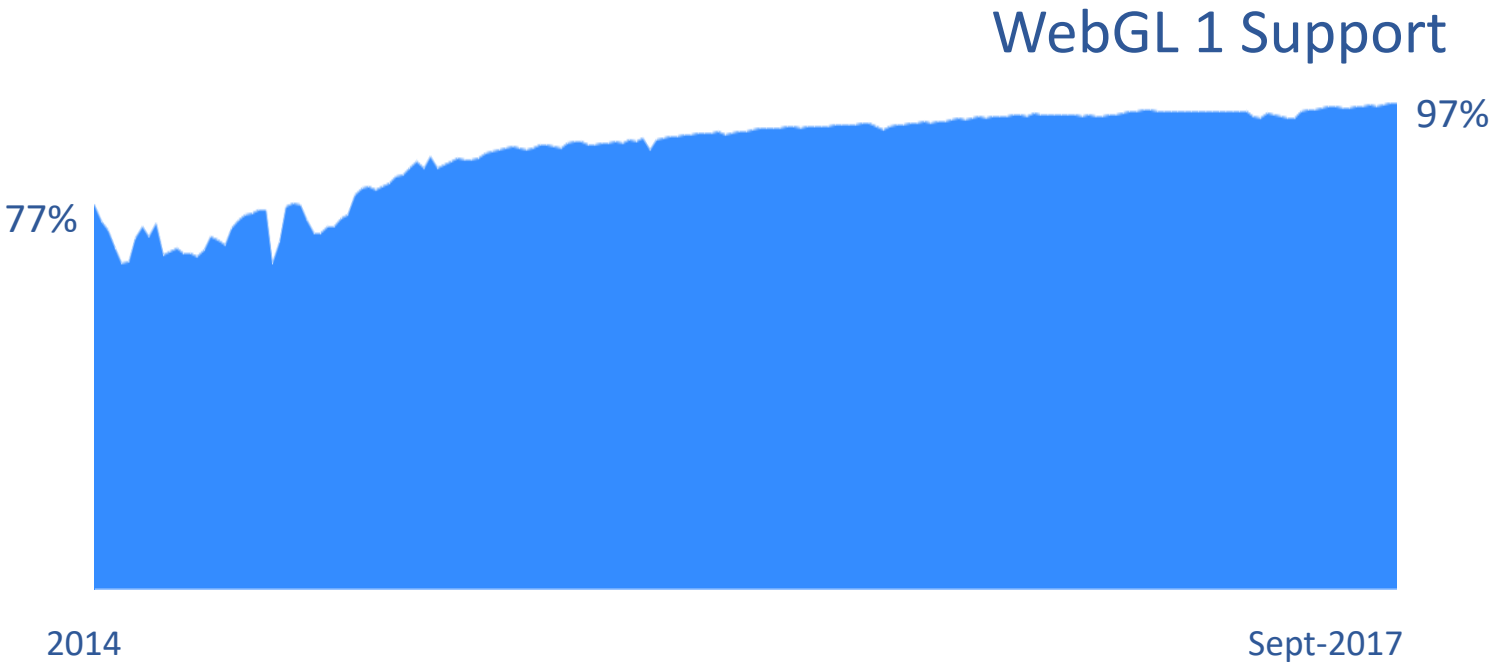
Updated: Thursday, 04 Feb 2016, 03:09PM

- Personal computer 88%
  - Windows 74%
  - OS X 14%
  - Chrome OS 10%
  - Linux 2%
  - Other 0.6%
- Smartphone 8%
  - Android 54%
  - iOS 44%
  - Windows 2%
  - Other 0.5%
  - Firefox OS 0.0%
- Tablet 4%
  - iOS 91%
  - Android 9%
  - Windows 0.1%
  - Other 0.0%
- Game console 0.1%
  - Playstation 47%
  - Other 25%
  - XBox 23%
  - Nintendo 5%
- Smart TV 0.0%
  - Linux 77%
  - Other 23%
  - Windows 0.0%
- PDA 0.0%
  - Palm OS NaN%

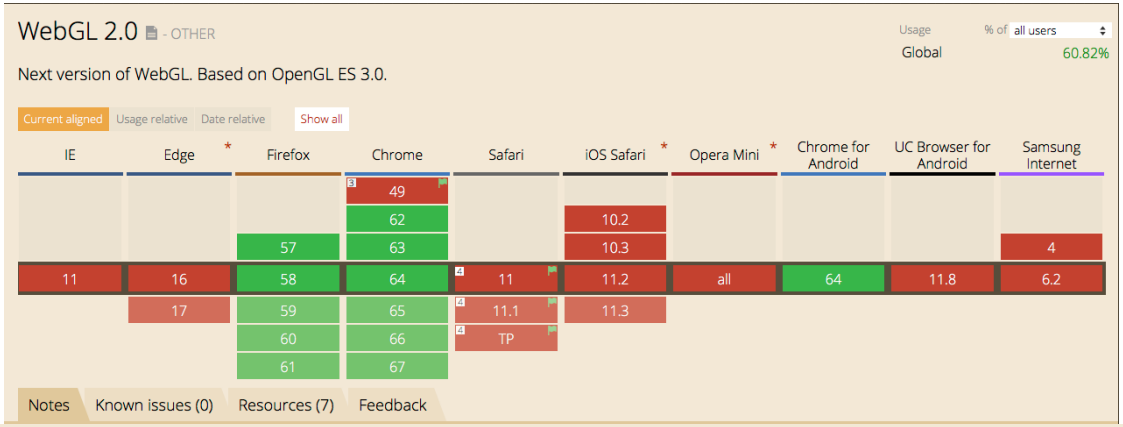


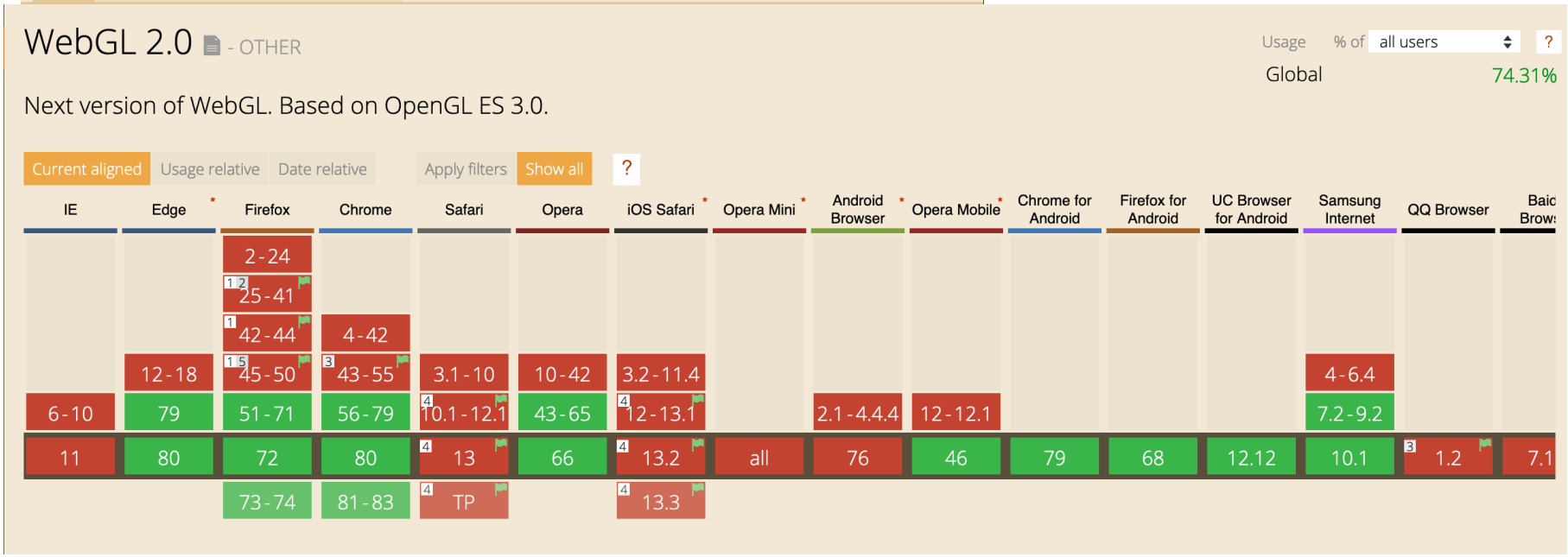


2017



https://webglstats.com/webgl


WebGL 2.0 - OTHER
Usage
% of all users
Global
60.82%
Next version of WebGL. Based on OpenGL ES 3.0.
Current aligned
Usage relative
Date relative
Show all
IE
Edge
Firefox
Chrome
Safari
iOS Safari
Opera Mini
Chrome for Android
UC Browser for Android
Samsung Internet
11
16
17
57
58
59
60
61
62
63
64
65
66
67
49
10.2
10.3
11
11.1
TP
11.2
11.3
all
64
11.8
4
6.2
Notes
Known issues (0)
Resources (7)
Feedback

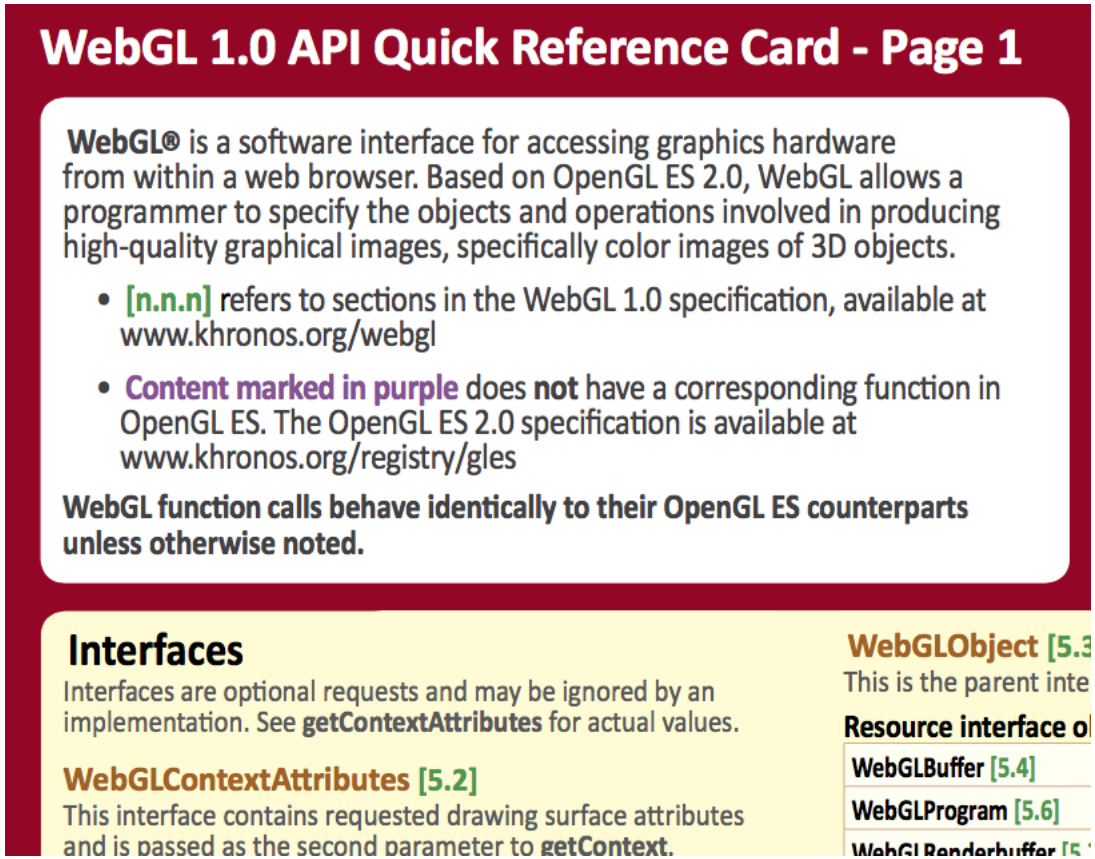

WebGL 2.0 - OTHER
Usage
% of all users
Global
74.31%
Next version of WebGL. Based on OpenGL ES 3.0.
Current aligned
Usage relative
Date relative
Apply filters
Show all
IE
Edge
Firefox
Chrome
Safari
Opera
iOS Safari
Opera Mini
Android Browser
Opera Mobile
Chrome for Android
Firefox for Android
UC Browser for Android
Samsung Internet
QQ Browser
6-10
11
12-18
79
80
2-24
25-41
42-44
45-50
51-71
72
73-74
4-42
43-55
56-79
80
81-83
3.1-10
10.1-12.1
13
TP
10-42
43-65
66
3.2-11.4
12-13.1
13.2
13.3
all
2.1-4.4.4
76
12-12.1
46
4-6.4
7.2-9.2
10.1
1.2
7.1

Demos

- https://www.khronos.org/registry/webgl/sdk/demos/google/shiny-teapot/index.html
- https://www.khronos.org/registry/webgl/sdk/demos/google/nvidia-vertex-buffer-object/index.html
- http://akirodic.com/p/jellyfish/
- https://www.shadertoy.com/
- http://madebyevan.com/webgl-water/
- https://webglsamples.org/

Reference Cards

- Auf Illias



WebGL 1.0 API Quick Reference Card - Page 1

WebGL® is a software interface for accessing graphics hardware from within a web browser. Based on OpenGL ES 2.0, WebGL allows a programmer to specify the objects and operations involved in producing high-quality graphical images, specifically color images of 3D objects.

- [n.n.n] refers to sections in the WebGL 1.0 specification, available at www.khronos.org/webgl
- Content marked in purple does **not** have a corresponding function in OpenGL ES. The OpenGL ES 2.0 specification is available at www.khronos.org/registry/gles

WebGL function calls behave identically to their OpenGL ES counterparts unless otherwise noted.

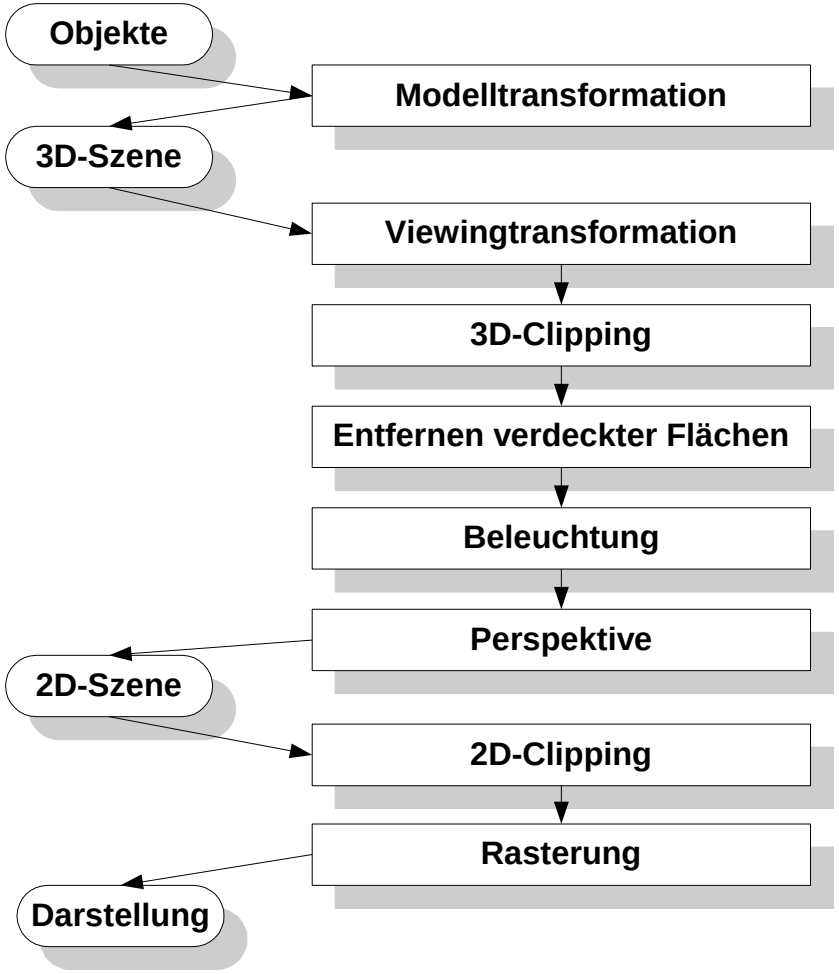
Interfaces

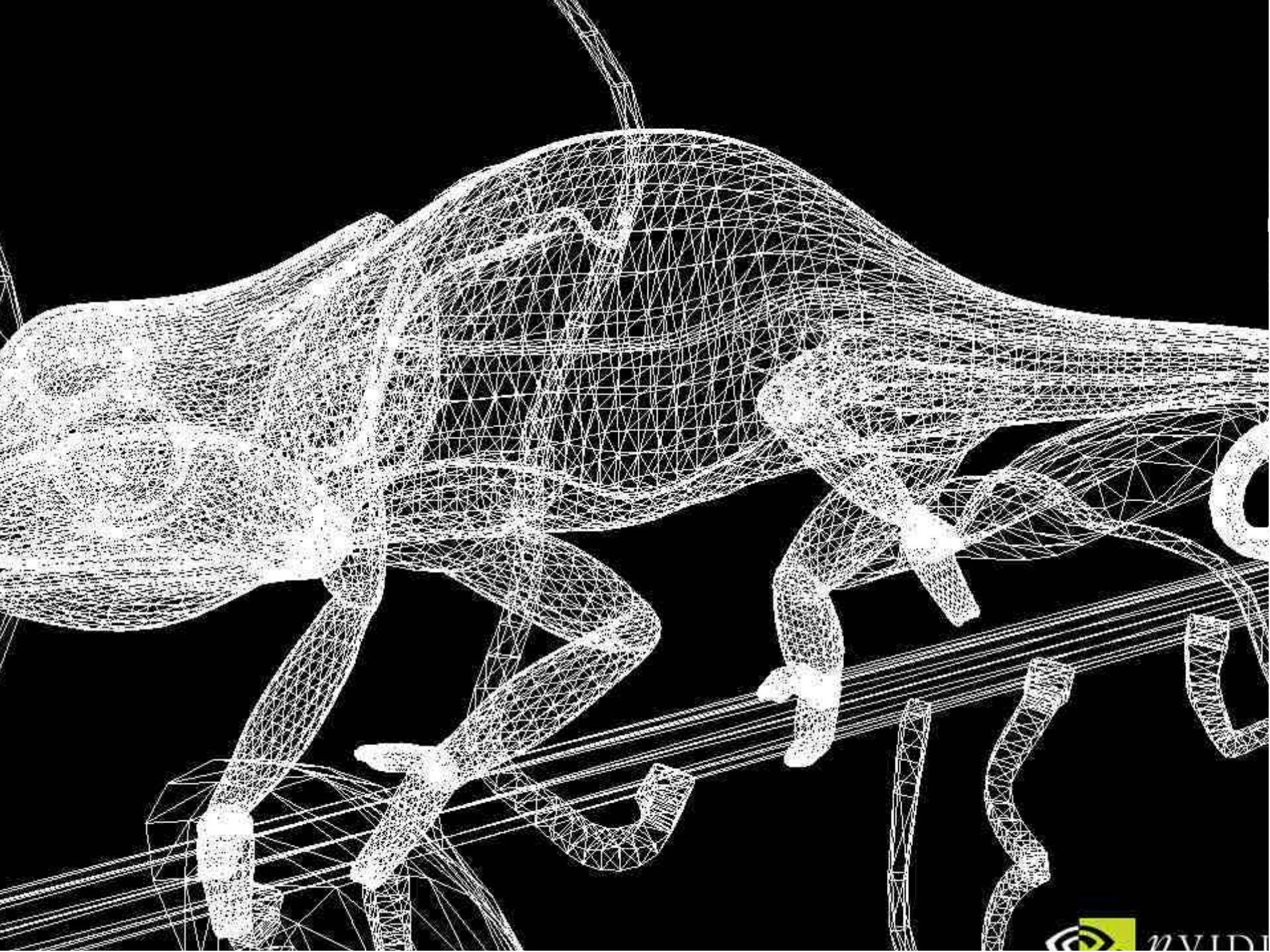
Interfaces are optional requests and may be ignored by an implementation. See **getContextAttributes** for actual values.

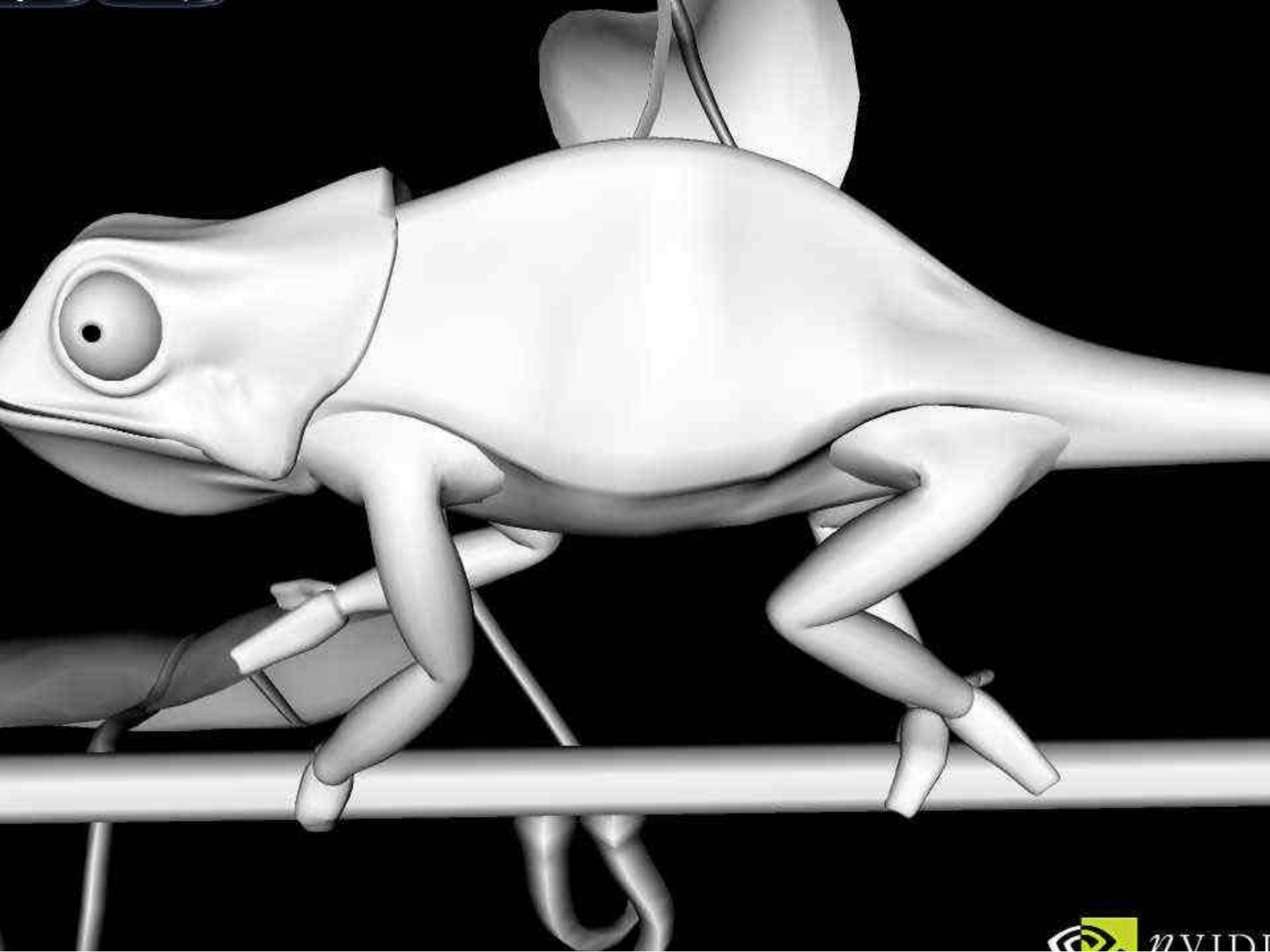
WebGLContextAttributes [5.2]
This interface contains requested drawing surface attributes and is passed as the second parameter to **getContext**.

WebGLObject [5.3
This is the parent inte
Resource interface o
WebGLBuffer [5.4]
WebGLProgram [5.6]
WebGLRenderbuffer [5

Grafik Pipeline zur 3D Darstellung







Prorammierbare Shaders

- Direkte Ausführung auf der Grafik-Hardware

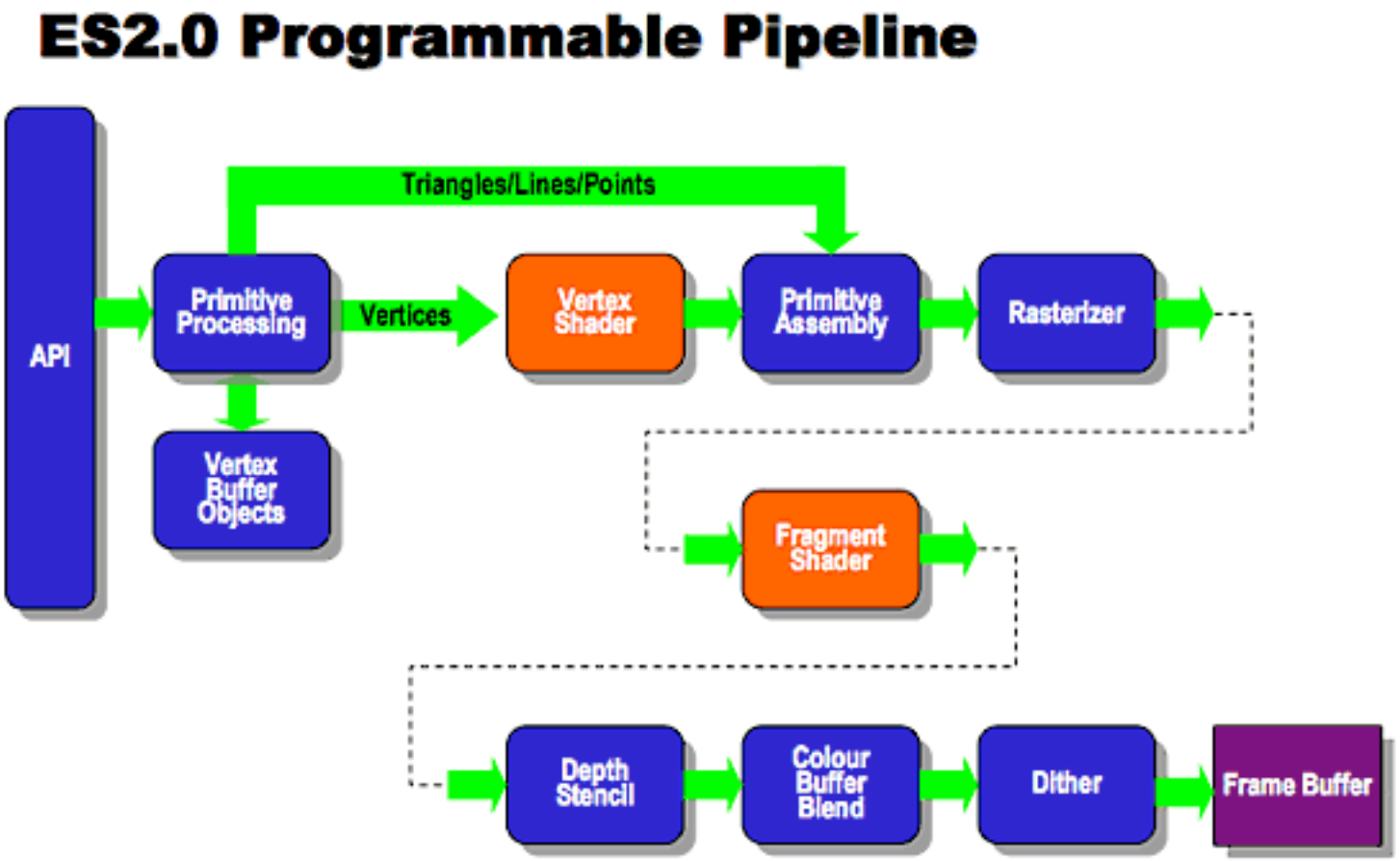
- Beispiel Bump mapping: Während der Berechung wird der Normalenvektor verändert

RenderMan: non-interactive Shader Languages

© Disney/Pixar

King Kong © 20th Century Fox

WebGL Pipeline



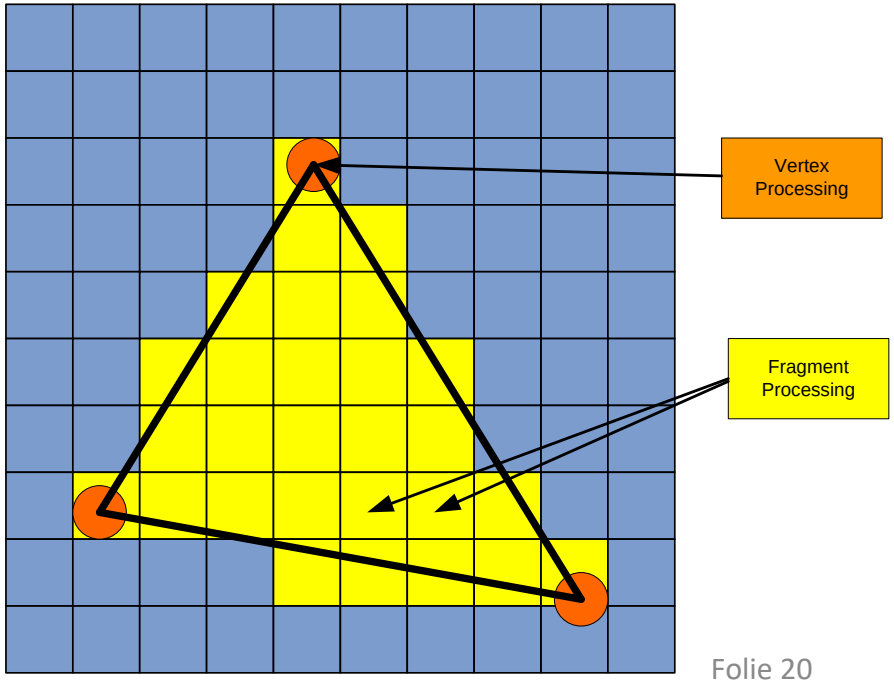
Vertex & Fragment Processing

Vertex Processing:

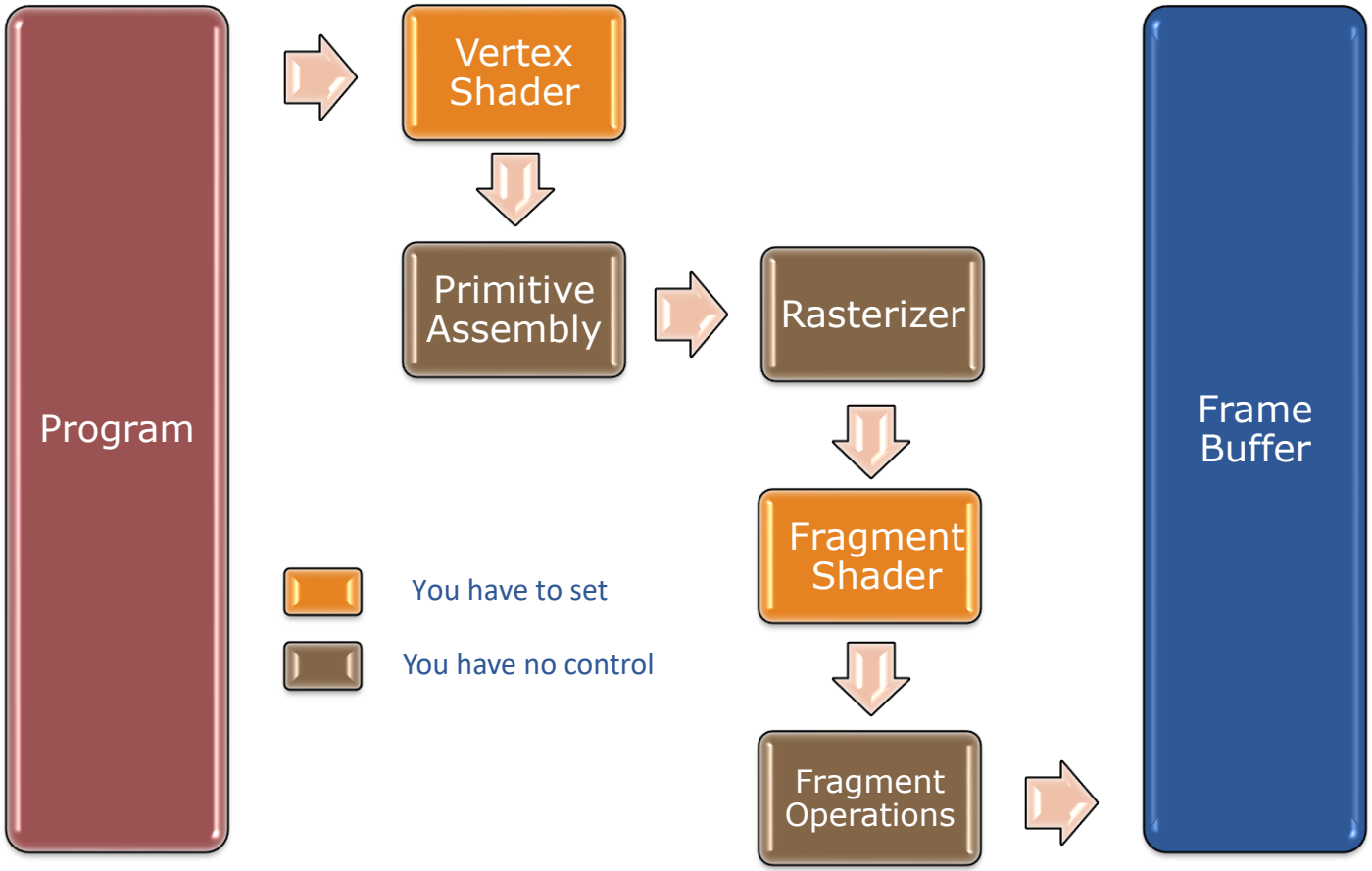
- Berechnet die Position eines Vertex,
- Vorberechnungen von weiteren Daten für Fragment Processing möglich

Fragment Processing:

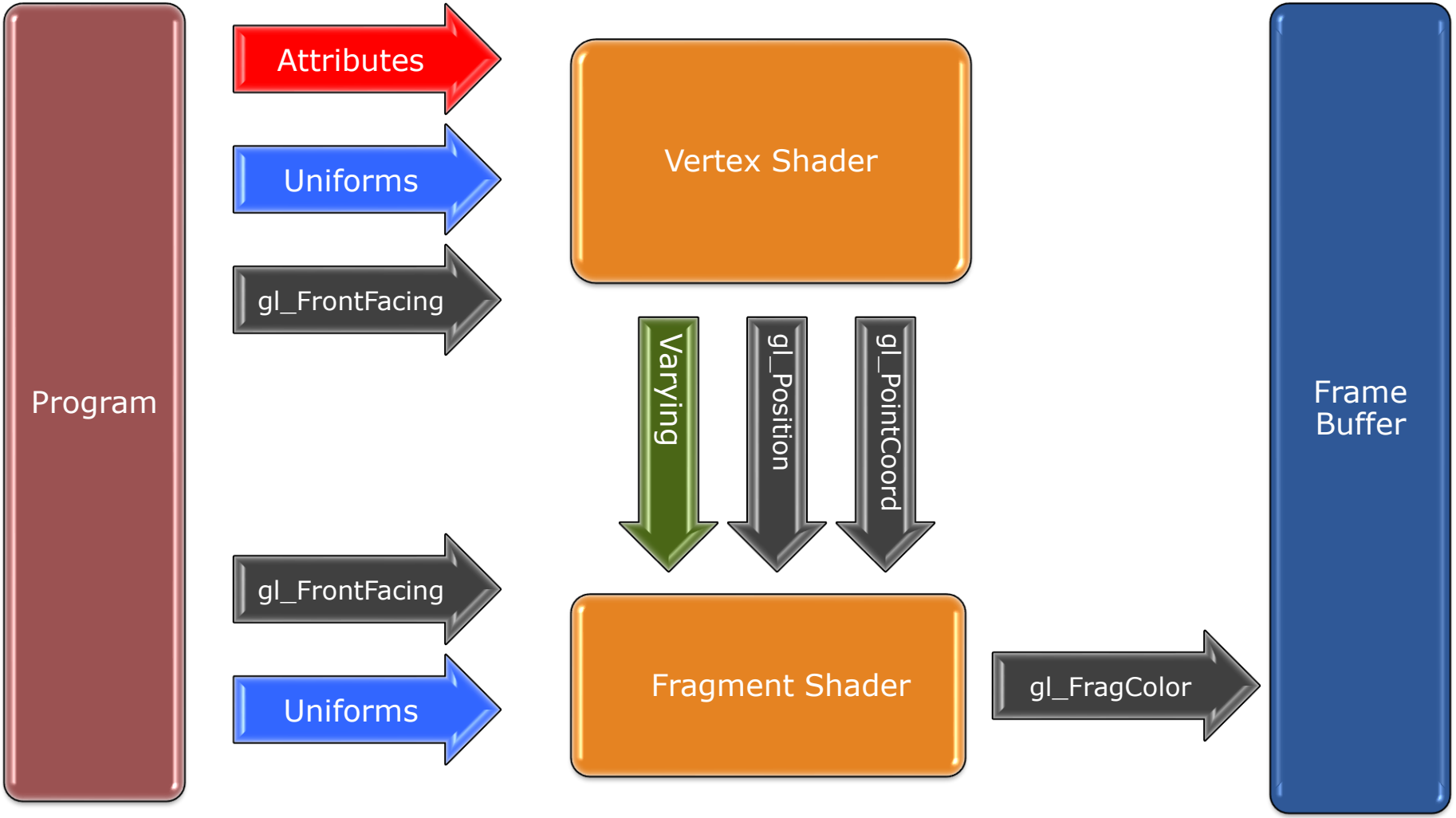
- Berechnet die Farbe des Pixels



WebGL Pipeline (simplified)



WebGL Data Flow



Attribute Variables und Buffer

Erzeugen

1. Buffer erzeugen (gl.createBuffer())
2. Array Buffer auf neuen Buffer setzen (gl.bindBuffer (gl.ARRAY_BUFFER, bufferId)
3. Daten füllen (gl.BufferData (…))

Zeichnen

1. bindBuffer (falls mehrere Buffer verwendet werden)
2. Attribute auf buffer setzen (gl.vertexAttribPointer (…)
3. (Enable Attribute als Array (gl.enableVertexAttribArray (…)
4. Zeichnen (gl.drawArrays (…))

Attribute Variables

getAttribLocation (program, DOMString name)

returns the generic attribute index that the attribute variable named name was bound to when the program object named program was last linked

Attribute Variables

void vertexAttribPointer (GLuint index, GLint size, GLenum type, GLboolean normalized, GLsizei stride, GLintptr offset)

Assign the WebGLBuffer object currently bound to the ARRAY_BUFFER target to the vertex attribute at the passed index. Size is number of components per attribute. Stride and offset are in units of bytes. Passed stride and offset must be appropriate for the passed type and size or an INVALID_OPERATION error will be generated; see Buffer Offset and Stride Requirements. If offset is negative, an INVALID_VALUE error will be generated. If no WebGLBuffer is bound to the ARRAY_BUFFER target, an INVALID_OPERATION error will be generated. …

WebGL Attrib

void vertexAttrib[1234]f (GLuint index, ...)

void vertexAttrib[1234]fv (GLuint index, ...)

Sets the vertex attribute at the passed index to the given constant value. ...

Uniform Variables

getUniformLocation (program, DOMString name)

Return a new WebGLUniformLocation that represents the location of a specific uniform variable within a program object. The return value is null if name does not correspond to an active uniform variable in the passed program.

Uniform Variables

void uniform[1234][fi] (location, ...)

void uniform[1234][fi]v (location, ...)

void uniformMatrix[234]fv (location, GLboolean transpose, ...)

Each of the uniform* functions above sets the specified uniform or uniforms to the values provided. If the passed location is not null and was not obtained from the currently used program via an earlier call to getUniformLocation, anINVALID_OPERATION error will be generated. If the passed location is null, the data passed in will be silently ignored and no uniform variables will be changed. …

Drawing primitives

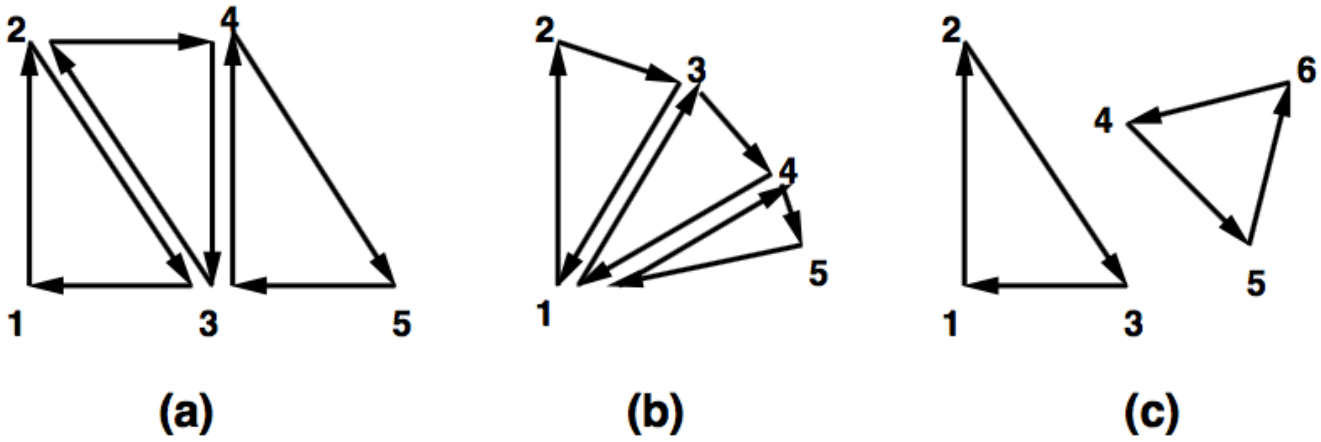


Figure 2.3. (a) A triangle strip. (b) A triangle fan. (c) Independent triangles. The numbers give the sequencing of the vertices in order within the vertex arrays. Note that in (a) and (b) triangle edge ordering is determined by the first triangle, while in (c) the order of each triangle's edges is independent of the other triangles.

Resources für WebGL/OpenGL ES

Programming Guide:

- https://sites.google.com/site/webglbook/

Specification:

- https://www.khronos.org/registry/webgl/specs/latest/

Manual pages for OpenGL ES

- https://www.khronos.org/opengles/sdk/docs/man/

Tutorials (many more)

- http://webglfundamentals.org/webgl/lessons/webgl-fundamentals.html
- http://math.hws.edu/graphicsbook/index.html