Portfolio

Copyright

Grundsätzlich ist das Material im Internet wie bspw. Bilder, Videos und Audios urheberrechtlich geschützt. Man kann sie also nicht ohne Bewilligung des Urhebers verändern, modifizieren oder republizieren. Aus diesem Grund ist, wenn man ein Bild auf der eigenen Website einfügen möchte, immer die Quelle anzugeben mit dem entsprechenden Hinweis auf die Wiederverwendung. Einfacher ist es jedoch Material zu verwenden, welches nicht urheberrechtlich geschütz ist. Dies sind beispielsweise die eigenen Werke oder Material, welches ausdrücklich zur wiederverwendung im Internet gekennzeichnet wurde.

Bilder

PPI

PPI steht für Pixels per Inch und gibt an, was für eine Pixeldichte das Bild hat. Je höher der Wert ist, desto höher ist die Auflösung des Bildes und desto meht Details können in den Bildinformationen gespeichert werden.

Dateiformate

FORMAT	VORTEILE	NACHTEILE
JPG	• Gute Komprimierung • Gut geeignet für Web, da kleinere Dateie	• Kein Alpha-Kanal (Keine Transparenz) • Keine Bewegtbilder

FORMAT	VORTEILE	NACHTEILE
PNG	• Unterstüzung für durchsichtige Bilder • Verlustfreie Kompression • Alpha-Transparenz in allen Browsern ausser bei dem Microsoft Internet Explorer 7 unterstützt	• Keine Bewegtbilder, bzw. nur mit MotionPNG • Keine Unterstützung für CMYK
GIF	• Transparenz • Animationenen	• Sehr alt • Eher grosse Datei, verglichen mit PNG

Mehr Informationen in Wikipedia

EXIF Informationen

JPEG bietet in ihren Metadaten sogenannte EXIF Daten an. Das sind Daten, wie der Kameratyp, der Objktivtyp, die Belichtungszeit, die Blendenöffnung, usw. Man kann diese Daten mit `exif_read_data` auslesen und dementsprechend verwenden. In meiner Galerie bspw. erscheint ein Pop-Up mit dem Kameratyp, wenn man mit der Maus drüberfährt.

Wasserzeichen

Das Wasserzeichen wird bei mir dynamisch in dem PHP Code hinzugefügt. Dazu waren einige Zeilen Code nötig:

▶ Code anzeigen

Dabei handelt es sich um ein sichtbares Wasserzeichen, welches in das Bild hineingefügt wird. Es gibt aber auch noch die Möglichkeit des unsichtbaren Wasserzeichen, welches ein lediglicher Eintrag in die Metadaten ist, sodass der Urheber mittels Software danach ausgelesen werden kann.

Verschiedene Präsentationsvarianten auf Webauftritten

Es gibt viele Methoden, um Bilder in Websiten zu integrieren. Die häufigsten sind jedoch das Karusell und die Galerie.

	Karusell	**Galerie**
Vorteile	• Eine Selektion von guten Bildern wird automatisch dargestellt • Lenkt die Aufmerksamkeit des Nutzers auf sich	• Kann eine grosse Menge an Bildern darstellen • Sehr übersichtlich

Nachteile	• Unübersichtlich • Nicht für viele Bilder geeignet	• Stellt Inhalte nicht so prominent dar • Kann viele Klicke erfordern, bis man am Ziel ist

Banner

Banner können beispielsweise für Werbung verwendet werden. Dabei haben sie meistens eine Standardgrösse. Eine Übersicht von Google Ad Words Banner Sizes findet man hier. Früher waren die Banner meistens mit Flash erstellt worden, um dynamische Inhalte zu erstellen. Jedoch ist Flash mittlerweile veraltet, weshalb die meisten Banner auf HTML5 mit einem Canvas umgestiegen sind.

JavaScript

Ich habe vor ca. 6 Jahren mit JavaScript begonnen und diverse Projekte vor dem Eintritt in die BBW erstellt. So habe ich das Teapot-Projekt schon vorher gemacht, da mich 3D Grafiken interresiert haben.

HTML5

Komptibilität

HTML5 wird heute von so ziemlich allen Browsern unterstützt. Jedoch unterstützen einige Browser nicht alle Tags restlos. Um herauszufinden, welche HTML5 Funktionalität wo unterstützt wird, finde ich die Site Can i use? sehr nützlich. Dort kann man einen Tag im Suchfeld eingeben und sieht unten, ab welcher Browserversion es unterstütz wird.

So kann man bspw. herausfinden, dass Audio Tracks in allen Browsern ausser Mozilla Firefox, Google Chrome für Desktop und Android und Opera unterstützt wird. Man kann aber auch gleich ganze Technologien auf Browserkompatibilität überpfüfen. So wird bspw. die Web Audio API von allen Brwosern, ausser dem Microsoft Internet Explorer unterstützt.

LocalStorage

In HTML5 gibt es die Möglichkeit, via LocalStorage über JavaScript Daten auf dem Client (wie Notizen) zu speichern. Dabei kann man mit `localStorage.setItem()` etwas speichern, und mit `localStorage.getItem()` den Eintrag wieder zurückbekommen.

CSS3

CSS3 wird nun von (fast) allen Browsern unterstützt. Sehr nützlich in CSS3 finde ich die `calc()` -Funktion, was Berechnungen wie `calc(50% - 3em + 20px)` ermöglicht.

Ich habe CSS3 in dem Projekt 3D Uhr verwendet, um HTML5 Elemente so zu transformieren, dass es so aussieht, als ob sie in einem 3D Raum fliegen würden. Ermöglicht wird dies durch die CSS3 Regel `-webkit-perspective` .

Canvas

WebGL

WebGL ist ein Bestandteil vieler moderner Webbrowser, mit dessen Hilfe 3D-Grafiken hardwarebeschleunigt ohne zusätzliche Erweiterungen dargestellt werden können
- Wikipedia

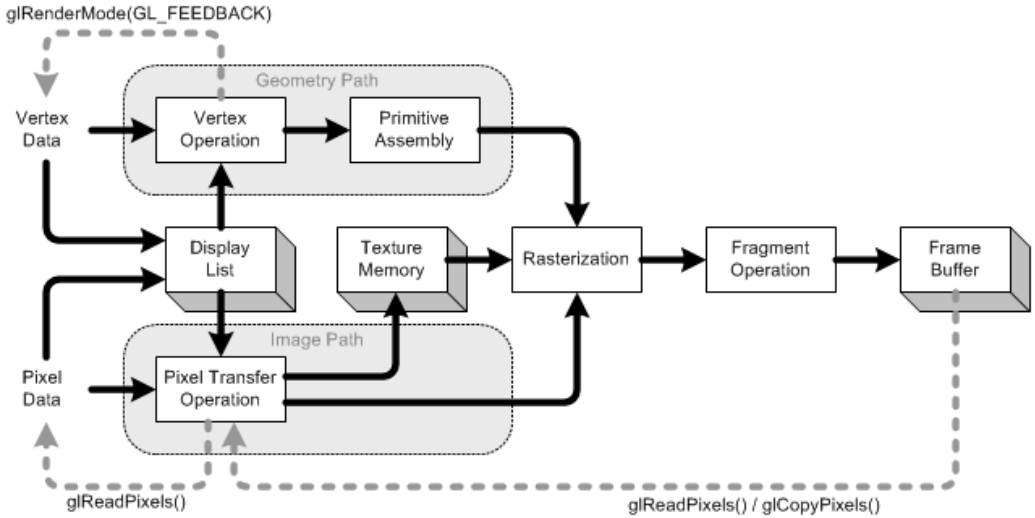
Funktionsweise

Um mit WebGL 3D Grafiken in ein Canvas zu rendern, braucht man diverse Elemente, welche in der folgenden Tabelle beschrieben werden.

ELEMENT	ERKLÄRUNG
Vertex-Shader	Dies ist der Code, welcher auf der Grafikkarte für jeder einerlne Koordinationpunkt des Modells ausgeführt wird. Er transformiert diesen Vektor dann so, dass eine 3D Darstellung entsteht und so, dass das Modell gemäss den Simuationen und User-Inputs affin transformiert wird. Vertex-Shader code
Fragment-Shader	Der Fragment-Shader ist ein Stück Code, das auf der Grafikkarte für jedes einzelne Pixel des Modells ausgeführt wird. Seine Aufgabe ist es, die Farbe des Pixels in Form eines vier-dimensionalen Vektors (r,g,b,a) zu berechnen. Der Fragment-Shader, welcher ich für den Teapot verwendet habe: Fragment-Shader code

ELEMENT	ERKLÄRUNG
Vertex-Buffer	Dieser Buffer enthält alle Positionsinformationen eines Modells. Er wird bei dem Rendern in einen Shared-Memory zwischen der Grafikkarte und der normalen CPU geladen, damit das Modell dann gerendert werden kann. Typischerweise werden in diesem Buffer auch noch die Normalen-Vektoren, die Indizes der einzelnen Koordinationvektoren, sowie die Texturkoordinaten übergeben. In meinem Beispiel lade ich jedoch die Normalen separat in einen zweiten Buffer, welche ich dann mit `attribute vec3 vertexNormals;` bekomme.
Program	Das Program ist ein OpenGL Objekt, welches den Vertex-Shader und den Fragment-Shader (in OpenGL 3+ auch den Geometry-Shader und den Tesselation-Shader) kompiliert hält. Es wird vor jedem Renderingaufruf verwendet, muss also jedes mal erneut gebunden werden, wenn das Programm vorher zu einem anderen gesetzt wurde.
Renderbuffer	In den Renderbuffer wird (Wie der Name schon sagt) hineingerendert. In OpenGL ES gibt es aber schon einen Standard-Buffer, weshalb man keinen eigenen machen muss, insofern man nicht offscreen rendert.
Framebuffer	Der Framebuffer enthält alle Informationen der Szene. Wenn man in den Hauptframebuffer rendert, dann wird dessen Inhalt auf dem Screen dargestellt.
Textur	Eine Textur ist eine ansammlung von verschiedenen Daten. Typischerweise wird ein Bild in eine Textur geladen, damit eine grosse Detailiertheit des Modells erreicht werden kann. Man kann aber auch in Texturen rendern, um später dann diese wieder in einem neuen Rendering-Call zu verwenden. So können bspw. Effekte wie Reflektionen realisiert werden.
Uniforms	Uniforms sind Variabeln, die man auf der CPU setzen kann, welche dann auf der GPU verfügbar sind. So kann man bspw. Lichtreflektionsinformationen eines Modells an die Grafikkarte, bzw. die Shaders, weitergeben, um das Licht zu berechnen. Mit dem Code `gl.uniform3f(program.vertexShaderAttributeLocations.diffuseLightColor, lightningOptions.point.R, lightningOptions.point.G, lightningOptions.point.B);` kann man bspw. die diffusen Lichtfarben an den Fragmentshader weitergeben.

Die Grafikpipeline



Quelle

Die oben dargestelle Grafik zeigt die Funktionsweise von OpenGL auf. Für eine detaillierte Erklärung verweise ich gerne auf diese Erklärung.

2D Kontext

Man kann auch in einem 2D Kontext in ein HTML5 Canvas rendern. Dazu wählt man einfach anstatt "experimental-webgl", den Kontext "2d" bei dem `canvas.getContext()` Aufruf aus. Die API sieht dann jedoch ein wenig anders aus. Man kann bspw. mit `ctx.lineTo();` eine Linie zeichnen, oder mit `ctx.arc();` einen Kreis.