**Digital Leaflet - Evaluation WebAR mit Three.js und AR.js**

Stand: 04.10.2018

Version: 1.0

# WebGL und Three.js

* JavaScript WebGL Framework
* Unterstützt werden folgende Browser (Minimalversion): Google Chrome 9+, Firefox 4+, Opera 15+, Safari 5.1+, Internet Explorer 11 and Microsoft Edge <https://threejs.org/docs/index.html#manual/en/introduction/Browser-support>
* Mobile Browser mit WebGL Unterstützung:
  + Safari on iOS 8
  + Opera Mobile 22+
  + Blackberry Browser 10
  + Chrome for Android 35 (Partial Support)
  + Firefox for Android 30 (Partial Support)

<https://caniuse.com/#feat=webgl>

* Grafische Qualität des Three.js Renderers ist sehr gut (Lighting, Physically Based Materials, Postprocessing / ImageEffects, …)
* FBX Importer (inkl. komplexer Animationen bis hin zu Skinned Mesh Animations)
* Soundschnittstelle (2D oder 3D Sound)
* 3D User Interfaces (3D Labels zur Anzeige von Komponentennamen am 3D Objekt)
* Weitere OpenSource Extensions und Module für Three.js vorhanden
* Ausführliche Dokumentationen, Beispiele und recht weit verbreitetes Framework

# AR.js

* JavaScript Framework mit Augmented Reality Funktionen für das Web
* Basiert auf einem JavaScript Port von ARToolkit (ca. 3 Jahre alt)
* Nachteile:
  + ARToolkit wurde 2015 von DAQRI aufgekauft und wird offiziell nicht mehr weiterentwickelt (letzte Version am May 13, 2015)
  + Keine detaillierte Dokumentation vorhanden
  + Tracker Erstellung ist eine Black Box (keine Dokumentation über optimales Trackerdesign vorhanden)
* Unterstützt werden alle Browser mit WebRTC und WebGL Unterstützung

<https://github.com/jeromeetienne/AR.js-docs/blob/master/misc/FAQ.md#which-plateform-is-supported>

* Lizenz: Open Source
  + jsartoolkit5 unter LGPLv3 Lizenz

<https://github.com/artoolkit/jsartoolkit5/blob/master/LICENSE.txt>

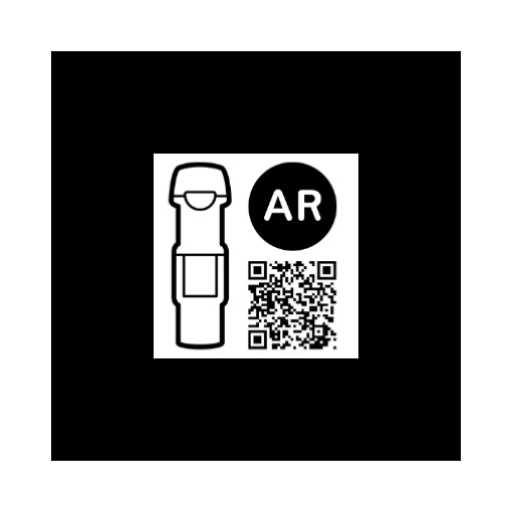
* + AR.js unter MIT Lizenz <https://github.com/jeromeetienne/AR.js/blob/master/LICENSE.txt>

# Testergebnisse

* Grundlage:
  + 2 Testszenen wurden erstellt
  + Einfache Szene mit Box zum Testen der Tracking-Stabilität:

<https://builds.realtimelabs.de/downloads/WebAR/simple.html>

* + Komplexe Szene mit animiertem Charakter zum Testen komplexer Geometrie (ca. 50.000 Polygone) in Kombination mit AR.js:  
    <https://builds.realtimelabs.de/downloads/WebAR/index.html>
  + Folgender Marker (5x5 cm) wurde für den Test verwendet (generiert mit AR Marker Generator  
    <https://jeromeetienne.github.io/AR.js/three.js/examples/marker-training/examples/generator.html> ):



* Ergebnisse:
  + Marker max. 80-100 cm erkennbar
  + Optimale Tracking Distanz: 10-50 cm
  + Jittering kann durch Implementierung eines Smoothing-Algorithmus (Interpolation der Kamera-Projektionsmatrix über mehrere Frames) noch weiter reduziert
  + Performance mit komplexer AR-Szene:
    - Samsung Galaxy S8: 50-55 FPS
    - iPhone 5s: 20-25 FPS
    - iPhone 8: 60 FPS
    - iPad 2017 und iPad Air 2: 30-35 FPS
    - Pixel C Tablet: 35-40 FPS

# Known Issues

* Android: Auf dem Samsung Galaxy S8 hatte der Samsung Browser Probleme bei der Fullscreen-Skalierung des Kamerabildes. Google Chrome hatte bei unseren Tests unter Android einwandfrei funktioniert.
* iOS: Auf iPhones hat die App im Safari Browser funktioniert. Beim Ausführen mit dem Chrome Browser unter iOS konnte die App nicht geladen werden.

# Ausblick

* Respimat 3D Modell:
  + CAD Modell: 475k Polygone
  + Echtzeit Modell sollte auf 5k bis max.10k Polygone reduziert werden
  + Optimierungen an Topology und Polygonanzahl sowie Erstellung Materialien und UV-Koordinaten für Texturierung des Modells notwendig
* WebXR:
  + Standardisierte Schnittstelle für VR und AR im Web (entwickelt u.a. von Mozilla, Google und Microsoft)
  + Wird zukünftig vrstl. der Standard für VR und AR im Web
  + Bisher noch kein öffentlicher Releasetermin angekündigt

<https://hacks.mozilla.org/2018/09/webxr/>

<https://developers.google.com/web/updates/2018/05/welcome-to-immersive>