

HTML 5-Editor für OpenStreetMap

Paul-Fiete Hartmann (OSM Benutzername: bastik)

21.3.2012

- ➊ Motivation
- ➋ Ähnliche Projekte
- ➌ Mein Prototyp (Beboj)
- ➍ Webtechniken (Details)

„HTML 5 Editor“?

- Läuft im Browser
- Benötigt keine Plugins / Laufzeitumgebungen
(wie Flash, Java, Silverlight)
- vollwertiger Editor für OpenStreetMap

Angepasste Versionen von Potlatch 2 auf verschiedenen Webseiten
(open.mapquest.co.uk, wanderreitkarte.de, skimap.org, usgs.com, ...)

Verschiedenste Webservices im OSM Umfeld

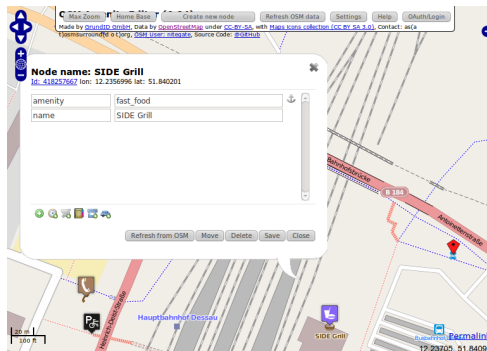
→ direkte und eng verknüpfte Editiermöglichkeiten, z.B.

- Qualitätssicherung (Keep Right, dupe nodes, ...)
- Kontrollierte Importe / Datenaktualisierung

Warum noch einen Editor?

- Verzicht auf Flash-Plugin
 - Engere Verzahnung mit anderen Funktionen der Webseite
 - Auf mobilen Geräten ohne Flash-Unterstützung einsetzbar
- Kompatibilität zu anderen (Javascript-)Bibliotheken (z.B. *Openlayers*)
- Mögliche „Wiederverwertung“ von JOSM Quelltext (s.u.)
- Technisch reizvoll

Ähnliche Projekte - Amenity Editor



von User:Nitegate

- Bearbeitung von POIs
- Ändern von Tags, Verschieben, Einfügen von Nodes
- (OAuth; Technische Umsetzung: Java Spring)

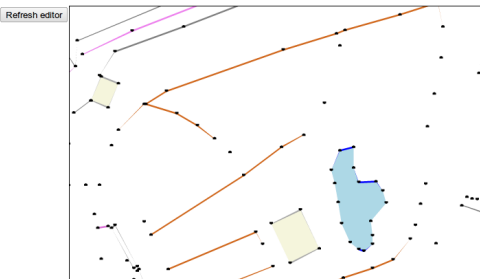
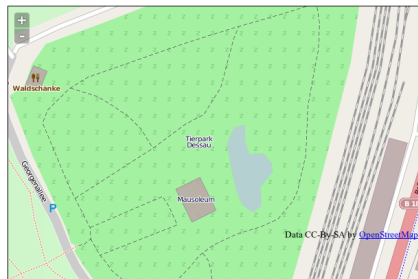
Ähnliche Projekte - GwtOsm



von James Michael DuPont (h4ck3rm1k3), 2010

- Nutzt JOSM+GWT
- SVG Rendering

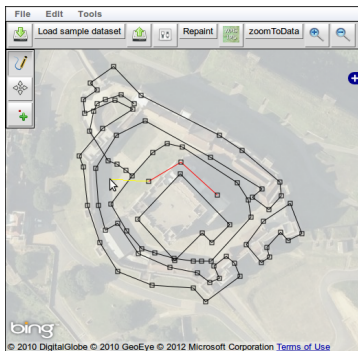
OpenStreetMap Editor



von Paul Wagener, 2010

- direkt in Javascript geschrieben
- Setzt auf SVG

Bejoj (Browser Editor based on JOSM)



Features:

- Download vom OSM-Server
- Maus-Interaktion im Potlatch 2 Stil
- Erstellen, Verschieben von Nodes und Ways
- Upload der Änderungen zum OSM-Server
- Bing-Luftbilder (auf Basis von Openlayers)

<https://github.com/openstreetmap/bejoj> (GPL 3+)

Welche HTML 5-Features sind essenziell?

Dynamisches Zeichnen / Rendering

- Notwendig für interaktives Editieren von Linien auf Clientseite
- Lösung: **Canvas** oder **SVG** (Scalable Vector Graphics)

Canvas

Zeichenfläche, zum Bemalen

Einfache low-Level Operationen

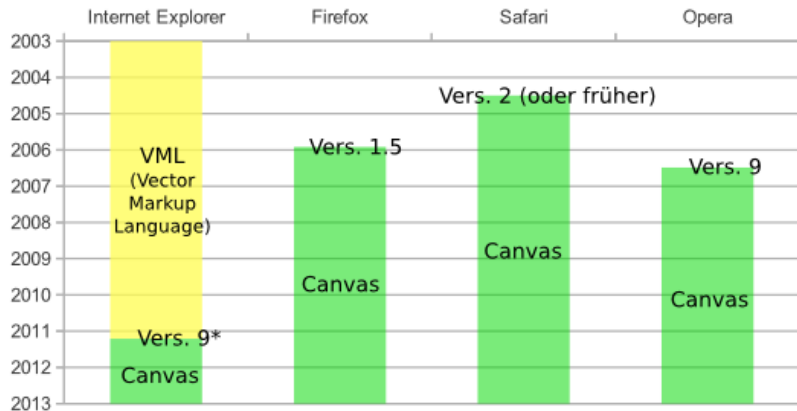
SVG

Beschreibungssprache für geometrische Formen

Reichhaltiger Standard, z.B.

- Eigenschaften per CSS
- Animationen

Canvas - Browserunterstützung



* nur für Win. Vista, Win. 7

Canvas - Codebeispiel



```
<canvas id="hot-canvas" width="70" height="40"></canvas>
```

```
<script type="text/javascript">  
  var example = document.getElementById('hot-canvas');  
  var c = example.getContext('2d');
```

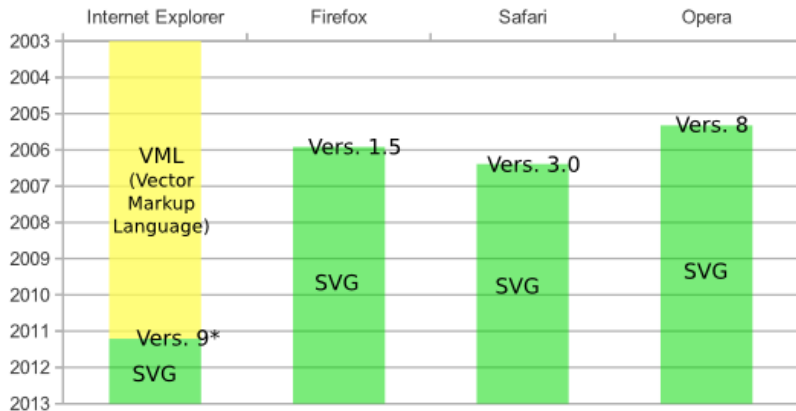
```
  c.beginPath();           /* Polygon Umriss */  
  c.fillStyle = '#d73f3f';  
  c.moveTo(10, 0);  
  c.lineTo(50, 0); /* ... */ c.lineTo(55, 20);  
  c.fill();
```

```
  c.beginPath();           /* Weisser Kreis */  
  c.fillStyle = 'white';  
  c.arc(35, 20, 13, 0, 2*Math.PI);  
  c.fill();
```

```
  c.fillStyle = '#d73f3f'; /* Text */  
  c.font = 'bold 16px sans-serif';  
  c.scale(0.5, 1);  
  c.fillText("HOT", 50, 25);
```

```
</script>
```

SVG - Browserunterstützung



* nur für Win. Vista, Win. 7

SVG - Codebeispiel



```
<svg width="70px" height="50px">
  <polygon
    points="10 0, 50 0, 70 20, 55 20, 65 30, 50 30,
           60 40, 20 40, 0 20, 15 20, 5 10, 20 10"
    fill="#d73f3f"/>

  <circle cx="35" cy="20" r="13" fill="white"/>

  <text x="50" y="25" textLength="40"
        style="font-size:16px; font-family: sans-serif;
              fill: #d73f3f; font-weight: bold"
        transform="scale(.5, 1)">HOT</text>
</svg>
```

Geolocation API Abfrage der **geographischen Ortsinformation** für Geräte mit GPS.

File API ¹ Zugriff auf **lokales Dateisystem**

- Gedacht z.B. für Vorschaubilder vor dem Upload von Fotos

Im OpenStreetMap-Umfeld:

- Fotomapping
- Laden von lokalen GPX-Dateien
- Zwischenspeichern ohne Upload

¹Experimentell, fehlende Unterstützung in wichtigen Browsern

Web Workers ¹ Lange **Hintergrundberechnungen**, ohne die Benutzeroberfläche zu blockieren.

Web Storage Ähnlich Cookies zum **Speichern von Daten im Webbrowser**.

(Aber deutlich mehr Speicherplatz)

- Speicherung von Einstellungen
- Speicherung von Ressourcen als Alternative zum Browser-Cache.

WebSockets ¹ **Bidirektionale Verbindung** zwischen einer Webanwendung und einem Server.

- Z.B. Kollaboratives Mapping in Echtzeit

¹Experimentell, fehlende Unterstützung in wichtigen Browsern

Google Web Toolkit (GWT)

Toolkit zur Entwicklung von Webanwendungen

ähnliche Frameworks:

- Apache Wicket
- Java Server Faces
- Java Spring
- ...

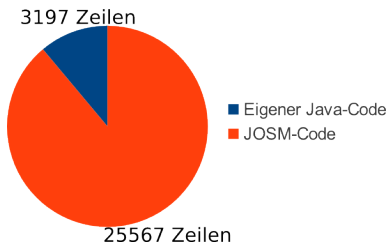
Alleinstellungsmerkmal von GWT:

Java-nach-Javascript-Compiler

→ Sowohl Server- als auch Client-Code kann in Java geschrieben werden

→ JOSM Code kann übersetzt und als Javascript im Browser ausgeführt werden

Beboj:



Google Web Toolkit (GWT)



Klappt so noch nicht vollständig: **Same-Origin-Policy**

Lösung:

Cross-Origin Resource Sharing (CORS)

- Server (OSM-Datenbank) muss mitspielen.

- HTML 5 Editor ist in der Entwicklung (soll keine Machbarkeitsstudie bleiben)
- Schlüsseltechnologien: Canvas & SVG
- Anregungen / Designvorschläge gern gesehen