Trailmap reittiyhteenveto ja -varoitukset

Trailmap v2 osaa tehdä reittianalyysin eli tarkistaa minkälaisia OSM-uria (way jolla highway tag) pitkin käyttäjän piirtämä tai kartalle muokkausta varten avattu reitti kulkee. Reittianalyysin pohjalta voidaan toteuttaa erilaisia toimintoja, aluksi ainakin:

1) Reittiyhteenveto eli yhteenveto kuljettujen urien valituista tag:eista. Aluksi highway ja surface tagit, mutta voidaan laajentaa kattamaan muitakin esim. Mtb:scale.
2) Reittivaroitukset eli käyttäjän varoittaminen automaattireitityksen valitsemista reitityksistä silloin kun ne ovat jollakin tapaa riskipitoisia. Tämän toiminnon avulla automaattireititys voi tehdä rohkeampia reittivalintoja ja käyttäjä sitten voi korjata tarpeen mukaan.

Reittivaroitukset

Reittivaroitukset pohjautuvat reititysprofiilikohtaisiin sääntöihin joita verrataan niiden urien tageihin joiden lävitse reitti kulkee. Jokaiselle reititysprofiilille luodaan omat säännöt.

Sääntöjen luomisessa tulee huomioida että urilla olevat tagit ovat vaihtelevia (eli ei voi olettaa, että esim. surface tagi olisi kaikilla teillä) ja niiden arvot eivät ole konsistentteja (esim. sorapintainen ulkoiluväylä voi olla track, cycleway tai jopa path).

Säännöt ovat muodoltaan Javascript objekteja joilla on seuraavat propertyt:

- title - Suomenkielinen varoitus joka näkyy käyttäjälle
- ruleType - enum jonka arvo voi olla WARNING tai ERROR eli varoitus (keltainen liputus) tai varma vaara (punainen liputus)
- matchType - POSITIVE tai NEGATIVE joka kertoo kuvaako sääntöfunktion tuottama “match” milloin liputus tapahtuu (POSITIVE) vai onko se poissulkeva eli jos “match” tapahtuu niin liputusta EI tapahdu (NEGATIVE)
- Matchfn - funktio jonka parametri on “tags”-muuttuja, joka on JS objekti joka sisältää arvioitavan uran tagit ja niiden arvot. Sääntö palauttaa JS boolean arvon.

ruleType on itseasiassa määritelty hieman hienostuneemmin, vaikka hienompia piirteitä ei vielä käytetä eikä ole loppuun asti mietitty:

```
export enum RuleType {
	WARNING = 'warning',
	ERROR = 'error',
	CONFIRMED_OK = 'confirmed_ok',
	CONFIRMED_ERROR = 'confirmed_error'
}
```

Ajatus on että säännöillä olisi seuraava prioriteetti: WARNING -> ERROR -> CONFIRMED_OK -> CONFIRMED_ERROR. Näistä CONFIRMED_OK on se “hienous” eli ajatus on, että sen tyyppinen sääntö yliajaisi WARNING ja ERROR säännöt, jollei sitten sitä yliaja

CONFIRMED_ERROR. Miksi ihmeessä? Mietitäänpä maantiepyöräily: haluamme antaa WARNING jos uralta puuttuu surface tagi, koska usein pienillä teillä se tarkoittaa että tietä ei ole pinnoitettu eli ei ole asfalttia. Mutta jos tie on iso “highway=trunk tai primary” on aika epätodennäköistä että tie olisi soraa vaikka surface tagi puuttuukin, silloin CONFIRMED_OK sääntö antaa mahdollisuuden yliajaa alemman prioriteetin WARNING ja ERROR säännöt. Tämän logiikan voi toki toteuttaa myös monimutkaisemmalla WARNING säännöllä - mikä toimii parhaiten selvinnee vasta kun lähtee sääntöjä tekemään ja kokeilemaan käytännössä oikealla OSM-datalla.

Tagien esi-prosessointi

Urien tagit esi-prosessoidaan ennen sääntöjä ja myös ennen reittiyhteenvedon laskemista. Esi-prosessoinnin tarkoitus on yhdistää erilaisia tageja niin, että suorien tagien sijaan käytetään niistä aggregoituja arvoja. Näin yhteenveto saadaan lyhyemmäksi ja varoituksien sääntöjen määrä tolkulliseksi.

Prosessointisäännöt ovat suoraviivaisia JS funktioita. Niiden kehittäminen pitää tapahtua yhdistettynä sääntöihin. Esim. tällä hetkellä access:bicycle tagi poistetaan esiprosessoinnissa jos se on “yes” tai “designated” - tämä on tarkoituksenmukaista koska useimmiten ajo on lähtökohtaisesti sallittu ja sääntöjen kannalta on kiinnostavaa vain jos ajaminen onkin kielletty. Mitä enemmän esiprosessoinnissa saadaan karsittua turhia tageja ja aggregoitua arvoja, sitä helpompaa on kehittää sääntöjä (toki kunhan tarpeellinen informaatio on tarjolla säännöille).

JS funktiot esi-prosessointiin alla - eivät vielä kata montaakaan tagia.

```
export function processHighwayTag(tagValue: string) {
	let result = null;

	switch (tagValue) {
		case 'path':
			result = 'path';
			break;
		case 'footway':
		case 'pedestrian':
			result = 'footway';
			break;
		case 'cycleway':
			result = 'cycleway';
			break;
		case 'track':
			result = 'track';
			break;
		case 'service':
		case 'residential':
```

```
                case 'unclassified':
                case 'tertiary':
                case 'living_street':
                case 'road':
                        result = 'minor_road';
                        break;
                default:
                        result = 'major_road';
        }

        return result;
}

export default function processSurfaceTag(tagValue: string) {
        let result = null;

        switch (tagValue) {
                case 'asphalt':
                        result = 'asphalt';
                        break;
                case 'paved':
                case 'concrete':
                        result = 'paved';
                        break;
                case 'compacted':
                case 'fine_gravel':
                        result = 'good_unpaved';
                        break;
                case 'unpaved':
                        result = 'unpaved';
                        break;
                case 'gravel':
                case 'pebblestone':
                        result = 'bad_unpaved';
                        break;
                case 'ground':
                case 'dirt':
                case 'earth':
                case 'grass':
                        result = 'ground';
                        break;
                default:
        }
```

```
	return result;
}

export function processBicycleTag(tagValue: string) {
	let result = null;

	switch (tagValue) {
		case 'yes':
		case 'designated':
			result = null;
			break;
		case 'no':
			result = 'no';
			break;
		default:
	}

	return result;
}

export function processMtbScaleTag(tagValue: string) {
	if (!tagValue || tagValue === '') return null;
	let result = tagValue;

	switch (tagValue) {
		case '0+':
			result = '0';
			break;
		case '1-':
		case '1+':
		case '-1':
			result = '1';
			break;
		case '6':
			result = '5';
			break;
		default:
	}

	return result;
}
```

Maantiepyörä säännöt

CONFIRMED_OK:

- highway=trunk | primary | secondary
- surface=asphalt

Huom.

- highway=tertiary tai alempi luokka voi olla unpaved tai vastaava

```
const roadbikeRules: PartitionRule[] = [
{
        title: 'Tien pinnoite tuntematon',
        ruleType: RuleType.WARNING,
        matchType: RuleMatchType.POSITIVE,
        matchFn: (tags: TAGS) =>
                !tags?.surface && !['trunk', 'primary', 'secondary'].includes(tags.highway)
},
{
        title: 'Tien pinnoite sora tai vastaava',
        ruleType: RuleType.WARNING,
        matchType: RuleMatchType.POSITIVE,
        matchFn: (tags: TAGS) => ['unpaved', 'good_unpaved'].includes(tags?.surface)
},
{
        title: 'Tien pinnoite karkea sora tai maa',
        ruleType: RuleType.ERROR,
        matchType: RuleMatchType.POSITIVE,
        matchFn: (tags: TAGS) => ['ground', 'bad_unpaved'].includes(tags?.surface)
},
{
        title: 'Polku tai maastotie',
        ruleType: RuleType.ERROR,
        matchType: RuleMatchType.POSITIVE,
        matchFn: (tags: TAGS) =>
                ['path', 'track'].includes(tags.highway) &&
                !(tags?.bicycle === 'designated') &&
                !['unpaved', 'good_unpaved'].includes(tags?.surface)
},
{
        title: 'Pyöräily kielletty',
        ruleType: RuleType.ERROR,
        matchType: RuleMatchType.POSITIVE,
```

```
            matchFn: (tags: TAGS) =>
                ['no', 'private'].includes(tags?.bicycle) ||
                ['no', 'private', 'permit'].includes(tags?.access)
        }
];
```

Gravel-pyörä säännöt

```
const gravelbikeRules: PartitionRule[] = [
        {
            title: 'Tien pinnoite tuntematon',
            ruleType: RuleType.WARNING,
            matchType: RuleMatchType.POSITIVE,
            matchFn: (tags: TAGS) =>
                !tags?.surface &&
                ['minor_road'].includes(tags.highway) &&
                !['0-', '0'].includes(tags['mtb:scale'])
        },
        {
            title: 'Tien pinnoite karkea sora tai maa',
            ruleType: RuleType.WARNING,
            matchType: RuleMatchType.POSITIVE,
            matchFn: (tags: TAGS) =>
                !['path'].includes(tags.highway) &&
                ['ground', 'bad_unpaved'].includes(tags?.surface) &&
                !['0-', '0'].includes(tags['mtb:scale'])
        },
        {
            title: 'Polku tai maastotie',
            ruleType: RuleType.WARNING,
            matchType: RuleMatchType.POSITIVE,
            matchFn: (tags: TAGS) =>
                ['path', 'track'].includes(tags.highway) &&
                !(tags?.bicycle === 'designated') &&
                !['unpaved', 'good_unpaved'].includes(tags?.surface) &&
                !tags['mtb:scale']
        },
        {
            title: 'Vaikea polku tai maastotie',
            ruleType: RuleType.ERROR,
            matchType: RuleMatchType.POSITIVE,
```

```
            matchFn: (tags: TAGS) =>
                ['path', 'track'].includes(tags.highway) &&
                !(tags?.bicycle === 'designated') &&
                !['paved', 'unpaved', 'good_unpaved'].includes(tags?.surface) &&
                !['0-', '0'].includes(tags['mtb:scale']) &&
                tags['mtb:scale']
        },
        {
            title: 'Märkä tai umpeenkasvanut ura',
            ruleType: RuleType.ERROR,
            matchType: RuleMatchType.POSITIVE,
            matchFn: (tags: TAGS) =>
                ['path', 'track'].includes(tags.highway) &&
                (tags?.obstacle === 'vegetation' || tags?.surface === 'mud')
        },
        {
            title: 'Pyöräily kielletty',
            ruleType: RuleType.ERROR,
            matchType: RuleMatchType.POSITIVE,
            matchFn: (tags: TAGS) =>
                ['no', 'private'].includes(tags?.bicycle) ||
                ['no', 'private', 'permit'].includes(tags?.access)
        }

];
```

Kommentteja muilta kirjoittajilta

Varoitus oranssista tai tiukemmasta polusta. Samoin vetinen tai puskittunut sekä hakkuut. Ehkä myös kapeimmasta keltaisesta voisi varoittaa, nämä yleensä ovat maastossa vaivoin erottuvia uria.

Ajokieltoalueille tuskin reititetään automaattisesti, mutta jos joku näin saa väkisin tehtyä, pitääkö tyylisyistä olla varoitus?

Muuta

- Jonkinlaisessa kartoittajanäkymässä voisi olla highlight teille joiden päällystettä ei ole määritelty