

Report Letny Semester

Ciele mojej práce v letnom semestri boli:

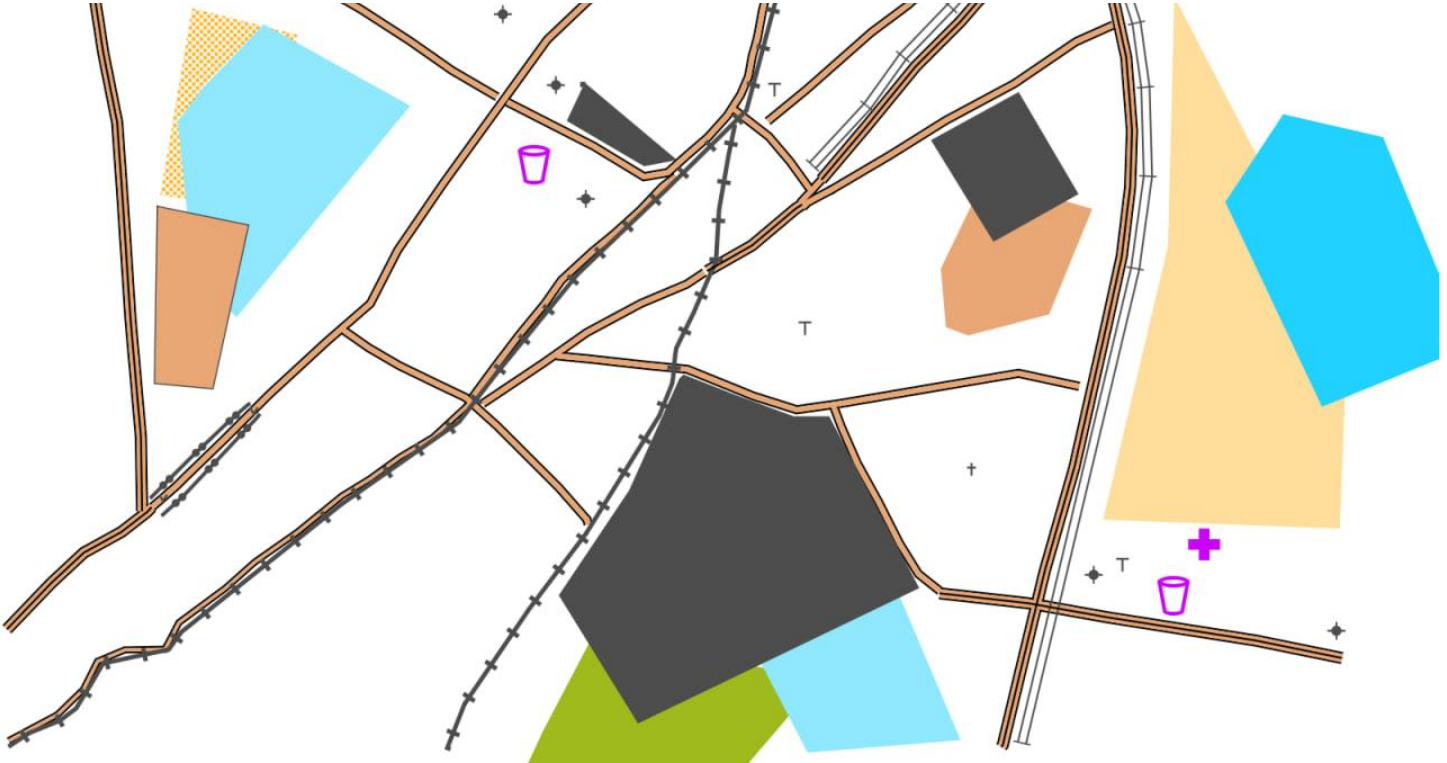
Identifikovanie prekryvy

Rozhodovanie o ich viditeľnosti

Pripadne - automatické odstránenie

Aby protestovať odstránenie prekryvy, urobil som doplnkovu mapu kde boli prekryvy na mape:

center_map_with_overlays.omap:



Po spustení Main.java tak to isto budeme vidieť mapy na vyber:

```
Select map to filter from the following list(1 - 3)
1. center_map.omap
2. center_map_with_overlays.omap
3. dedina_map.omap
2
```

Potom ukažeme, či chceme aby program niečo robil s prekryvami a čo s nimi robiť:

```
Do you want to handle overlapping objects? (y/n)
y
How do you want to handle overlapping objects?
1. Automatically remove lower priority objects
1|
```

Ďalej takto isto budeme mať možnosť ofiltrovať nejaké prvky z mapy:

```
Elements that were filtered the most in the center_map_with_overlays.omap
Map selected: center_map_with_overlays.omap
[Forest: reduced visibility, Building, Pond, Major power line]

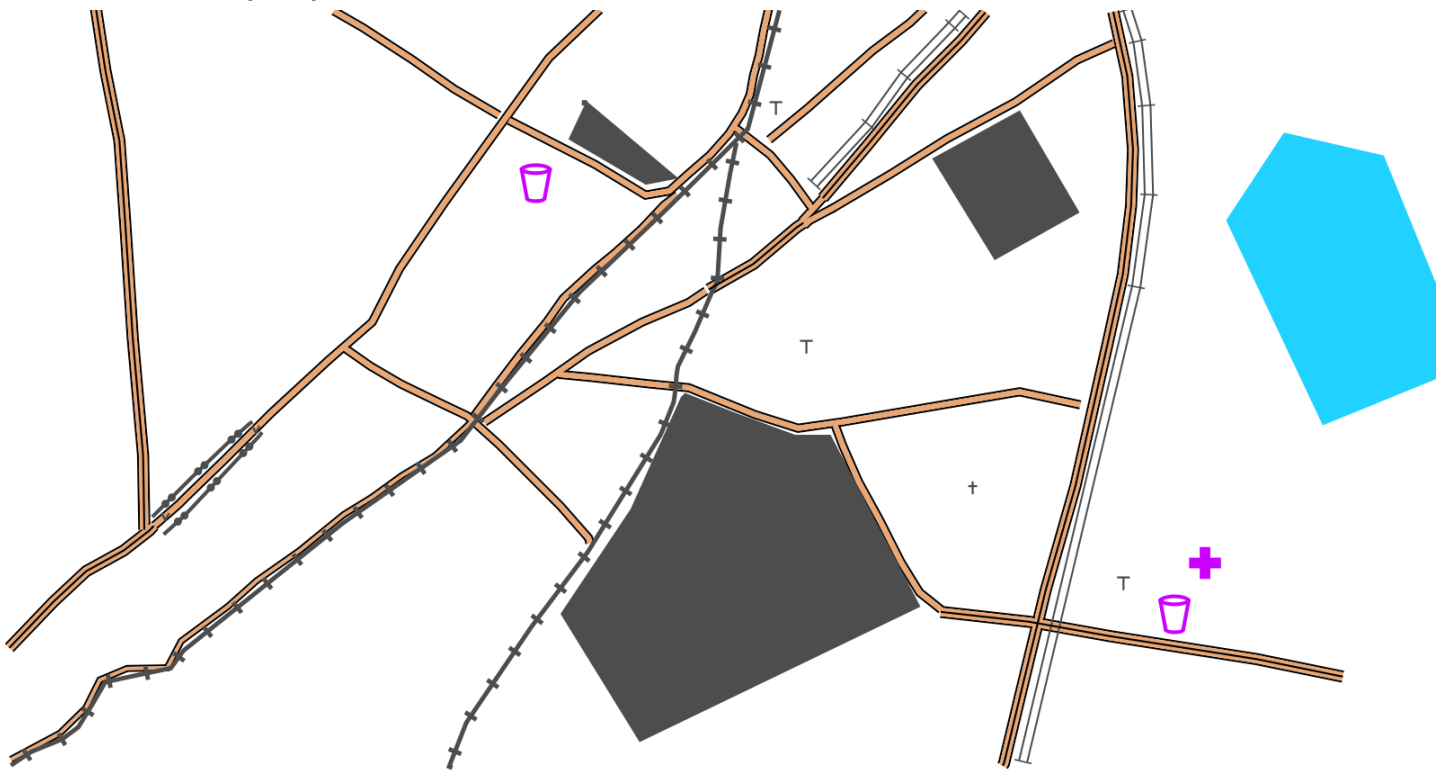
Enter object names to filter from the map (one per line, type 'exit' when done):
High tower
exit
```

Po dokončení budeme mať možnosť pozrieť si štatistiku:

```
Found 673 overlapping objects
Found 673 overlapping objects
Filtering complete!
Objects deleted from the map: 6
Overlapping objects handled: 7

Result saved to: RP\src\maps\custom_map_no_overlays.omap
```

A budeme mať takýto výsledok:



Cesty, káble a podobné veci som na mape nechal, pretože môžu prekryvať veľa krát, ostatným urobil som nejakú prioritu pre každý "symbol":

```

private int calculatePriority(Element symbol, String name) { 1 usage  ⚡ Maksym
    int priority = 50;
    name = name.toLowerCase();

    if (name.contains("building") || name.contains("tower")) {
        priority = 90;
    } else if (name.contains("water") || name.contains("lake") || name.contains("pond")) {
        priority = 80;
    } else if (name.contains("path") || name.contains("road") || name.contains("track") || name.contains("area"))
        priority = 85;
    } else if (name.contains("forest") || name.contains("vegetation") || name.equals("vineyard")) {
        priority = 30;
    } else if (name.contains("open") || name.contains("field")) {
        priority = 20;
    } else if (name.contains("contour") || name.contains("slope")) {
        priority = 70;
    } else if (name.contains("index")) {
        priority = 75;
    }

    if (symbol.getElementsByTagName("point_symbol").getLength() > 0 ||
        symbol.getElementsByTagName("point").getLength() > 0) {
        priority += 5;
    }

    if (symbol.getElementsByTagName("line_symbol").getLength() > 0 ||
        symbol.getElementsByTagName("line").getLength() > 0) {
        priority += 3;
    }
}

```

Pracu urobil: Maksym Pavlii

Mail: pavlii1@uniba.sk

Skupina: 2INF2

Školiteľ: Robert Lukotka

Mail Školiteľa: lukotka@dcs.fmph.uniba.sk