

# TBA4251

Programmering i Geomatikk

<b>Bakgrunn for oppgaven som er gjort</b>	<b>3</b>
<b>Formålet med applikasjonen</b>	<b>3</b>
<b>Valg av programmeringsspråk</b>	<b>3</b>
<b>Oppsummering av arbeidsgangen</b>	<b>3</b>
<b>Strukturen på programmet</b>	<b>4</b>
<b>Diskusjon</b>	<b>5</b>
Problemer som dukket opp underveis	5
Mangler og forbedringspotensialer	6
Hvor godt løser applikasjonen oppgaven	6
<b>Brukergrensesnittet</b>	<b>7</b>
<b>Referansar</b>	<b>7</b>
<b>Tutorial</b>	<b>7</b>
Bruk av programmet	8
Geografisk analyse	9

## Bakgrunn for oppgaven som er gjort

Oppgaven er gjort som en løsning på faget TBA4251. I dette faget kan man velge oppgave innenfor Geografisk Informasjonssystem (GIS) hvor oppgaven går ut på å lage sitt eget GIS.

## Formålet med applikasjonen

Formålet med applikasjonen er at den skal kunne opptre som en intro til GIS og bruken av GIS. GIS kan være avanserte og tidkrevende å sette seg inn og bruke. Min applikasjon skal være enkel å sette seg inn i og bruke og kan da brukes som en introduksjon til GIS.

Applikasjonen kan da brukes til å lære enkle GIS funksjonaliteter og bli kjent med måtene en kan bruke og utnytte funksjonaliteten i for eksempel ArcGIS eller QGIS.

## Valg av programmeringsspråk

Prosjektet baserer seg på React rammeverket. React er et rammeverk for JavaScript som er utviklet av Facebook. Rammeverket kombinerer HTML og javascript for å lage en mer strømlinjeformet måte å utvikle nettsider på. React er et rammeverk jeg har brukt i flere fag tidligere. I tillegg er det noe jeg har brukt i internship. Dette gjør at React er et naturlig rammeverk for meg og velge ettersom jeg kan det godt fra før og det er lagd spesielt til moderne webutvikling. React er kanskje verdens mest brukte rammeverk for webutvikling og det finnes fullstendig dokumentasjon og utallige spørsmål og svar på forumer dette er også en viktig grunn til at jeg valgte akkurat React.

For å visualisere kart og kartlag har jeg brukt Leaflet. Leaflet er et rammeverk laget for javascript for interaktive kart. All håndtering av kart og lagene bruker deler av Leaflet. Rammeverket er mye brukt i utvikling av kartapplikasjoner og har mye ressurser tilgjengelig. Leaflet er laget for JavaScript med et lite tillegg for React. Dette gjør at det ikke er like mye ressurser tilgjengelig på nettet. Noe som skapte litt problemer for meg som jeg vil nevne senere.

For GIS-analysen har jeg brukt et bibliotek som kalles Turf.js. Dette er et tillegg i JavaScript som hjelper med geografisk analyse. Jeg hadde ingen preferanser for dette på forhånd, men valgte Turf.js fordi det var mye brukt og hadde all funksjonalitet jeg var ute etter.

## Oppsummering av arbeidsgangen

Før jeg startet med å kode noe som helst tenkte jeg for meg selv hva jeg ville ha med på web-applikasjonen. Jeg tittet på gamle oppgaver levert av tidligere studenter og tenkte tilbake fra tidligere fag hvordan det var å bruke både ArcGIS og QGIS. Etter dette satte jeg opp en liste med ting jeg ville ha med i programmet.

- Mulighet til å gjennomføre grunnleggende GIS operasjoner
- Mulighet til å laste opp egen data
- Mulighet til å endre på rekkefølgen som lagene vises
- Mulighet til å endre fargen på lagene
- Mulighet til å velge om et lag skal synes eller ikke
- Mulighet til å slette et lag

Etter at jeg hadde laget en liste over funksjonaliteter jeg mente måtte være med i applikasjonen begynte jeg å kode.

Det første jeg gjorde var å sette opp utviklingsmiljøet. Deretter laget jeg en helt enkelt nettside for å sjekke at alt var OK. Videre begynte jeg å lage det grunnleggende på siden. Det første målet var å lage en nettside bestående av et kart med en kolonne ved siden av som kunne brukes til menyer osv. senere. Allerede her måtte jeg begynne å lese meg opp på nye ting. Jeg var ny med Leaflet og måtte da lære meg å bruke dette. Jeg liker når jeg utvikler å kode parallelt med å lære meg det. Dette er den beste måten for meg å lære meg nye rammeverk på. Derfor har jeg hele tiden lest meg opp på dokumentasjon samtidig som jeg har utviklet.

Etter dette var gjennomført gikk jeg over til å lage et system for å laste opp egne lag. Dette la jeg til i menyen til venstre på siden. Samtidig som dette ble gjort var det flere ting som måtte gjøres bak det som sees. Programmet måtte ha en måte å håndtere and distribuere de forskjellige lagene som blir lastet opp. Dette vil forklares mer senere i strukturen.

Etter dette begynte jeg på å lage muligheten til endre rekkefølgen på laget. Dette vil bli gått mye dypere inn i senere. Men her måtte jeg trikse og mikse mye for å få det til. Løsningen baserer seg på en *drag&drop* liste hvor man kan omrokkere rekkefølgen på lagene. Dette vil gjenspeiles på kartet.

Etter dette var gjort gikk jeg videre til å gjøre det mulig å slette lagene. Dette var noe som gikk ganske kjapt. Her var det som tok lengst tid å lage en knapp som fungerte på hensiktsmessig måte. Dette ble fulgt av at jeg lagde knapp for å endre på farge og gjennomsiktighet. I starten var denne funksjonen ganske minimal, men ble på et senere tidspunkt utviklet videre.

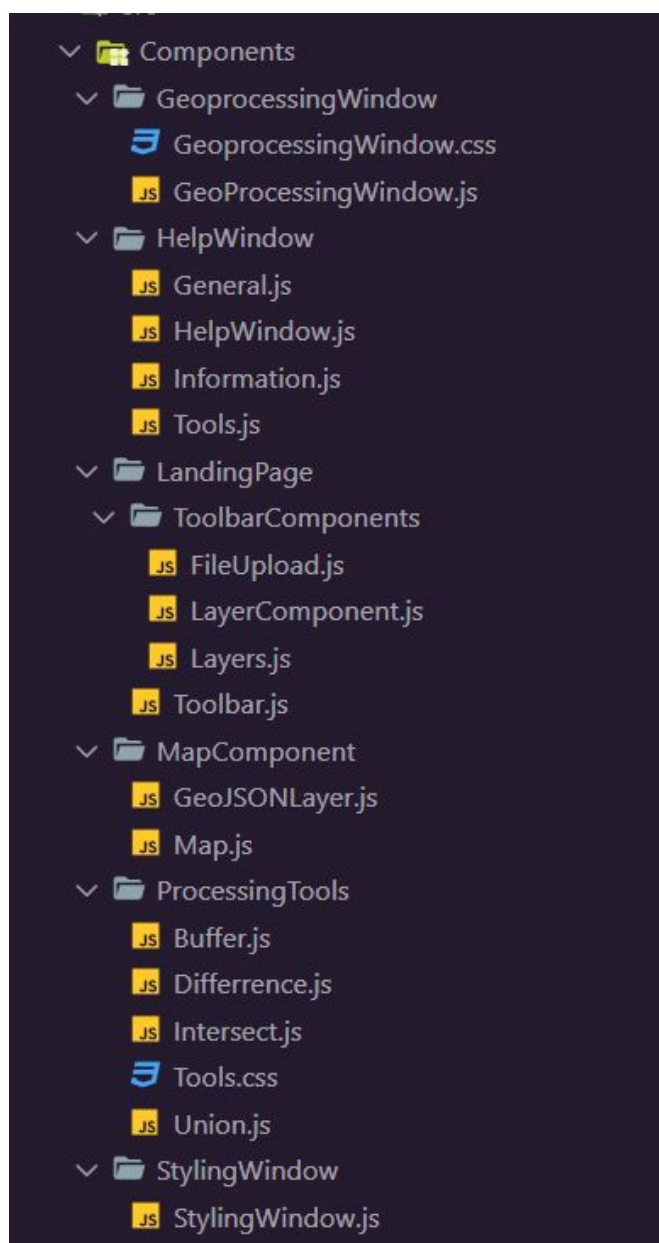
Da den grunnleggende funksjonaliteten til lagene var implementert gikk jeg over til å lage GIS funksjonalitetene. Grensesnittet de ble implementert med i starten var mye enklere og kun for å teste funksjonalitet. Dette prinsippet ble fulgt i hele utviklingen. Ettersom det var viktigere å implementere funksjonalitet enn et fint grensesnitt.

Da jeg var fornøyd med funksjonaliteten til gis-verktøyene gikk jeg over til å lage et bra grensesnitt. Dette ble gjort basert på tilbakemeldinger fra noen venner jeg viste nettsiden til. Dette førte til at jeg lagde et pop-up vindu som inneholder all GIS-funksjonalitet. Diverse knapper har blitt endret slik at de ser finere ut og like hverandre også.

Til slutt har jeg brukt tid på å finkjemme nettsiden for bugs og feil. Jeg har testet programmet godt og prøvd å lage feil. Dette har ført til at jeg har funnet ting som jeg enten har fikset eller lagd til instruksjoner for.

## Strukturen på programmet

Strukturen på programmet skiller seg lite fra vanlig koding i React. Koden består av et flertall komponenter som utgjør elementer på siden. For eksempel så er området med kart en komponent og hvert av lagene som vises er en komponent oppå kartet. Som tradisjonell HTML kode er det et hierarki som i vanlig XML som forteller hva som innkapsler hva.



## Diskusjon

### Problemer som dukket opp underveis

Det dukket opp flere problemer underveis i oppgaven. De aller fleste av de var relativt små som ikke tok alt for lang tid å fikse. Allikevel var det et problem jeg brukte flere uker på å finne en løsning på og vurderte til og meg og starte helt på nytt med nye rammeverk for å fikse. Dette var å implementere muligheten til å omrokkere rekkefølgen på lagene på kartet. Grunnen til at problemet oppstod i utgangspunktet var fordi leaflet ikke har en innebygd funksjon for å endre z-index til polygoner. GeoJSON som er formatet jeg bruker består av polygoner så da startet problemet. Jeg prøvde veldig mye forskjellig og vurderte å gå helt bort i fra Leaflet men etter mange timer og forsøk fant jeg en løsning som fungerte.

Måten jeg løste det på var å pakke polygon laget inn i en `<Pane />` laget. Dette gjorde at programmet ikke oppfattet kartlaget som et polygon, men som et Pane. Jeg vet ikke hvor robust denne løsningen er, men har ikke sett noe problemer med den til nå. Siden det ikke eksisterer noe andre løsninger på problemet, i hvertfall som jeg har funnet, ble jeg utrolig fornøyd da jeg fikk det til. Det kan virke som det er en ting de har oversett når de utviklet verktøyet fordi det er ekstremt nyttig.

Et annet problem jeg brukte en del tid på å fikse oppstod da jeg la til muligheten til å slette lag. Da oppstod et problem som gjorde at navnet på laget som ble vist ikke alltid stemte overens med laget som laget som ble vist. Dette gjorde det helt umulig å holde styr på noe som helst og ødela hele flyten i programmet. Dette problemet var enklere å løse men det tok en del tid før jeg fant frem til hvor problemet lå i koden. Da jeg fant problemet var det enkelt å fikse og ting fungerte som det skulle igjen.

Da noen av problemene oppstod var det flere ganger jeg vurderte å ikke bruke Leaflet lenger. Jeg begynte fort å bruke Leaflet uten å gjøre noe ordentlig undersøkelse om det var noen andre alternativer. Dette endte opp med å fungere, men jeg tvilte flere ganger på valget. Jeg valgte kun på bakgrunn av at Leaflet var mest brukt, men feilet med å sjekke nøye. Mapbox er et verktøy som er bygget oppå Leaflet og optimalisert for React. Hadde jeg skulle valgt på nytt hadde jeg brukt Mapbox i stedet. Dette er noe jeg tar lærdom av selv om det endte opp med å løse seg denne gangen.

## Mangler og forbedringspotensialer

For å gjøre applikasjonen enda bedre er det flere ting jeg kunne ha lagt til. Det er i hovedsak to ting jeg vil fokusere på så vil jeg legge til en liste med ting jeg kunne gjort i tillegg, men som er ikke annet enn "Quality of Life" oppdateringer.

Det første jeg gjerne skulle gjort er å gjøre det mulig å bruke lag som består av et flertall av polygoner. Dette er en ting jeg måtte unngå å implementere fordi det ville tatt for lang tid for meg. Noen av de gis verktøyene jeg bruker er ikke lagd for automatisk gjøres på lag med flere polygoner. Dette er noe jeg kunne implementert selv, men valgte å ikke gjøre for å spare tid. Dette er noe som kunne gjort at applikasjonen kunne vært brukt til enda mere. Jeg valgte å nedprioritere det ettersom applikasjonen kun måtte kunne håndtere forhåndspreparert data og at mangelen ikke hindrer demonstrasjon av verktøyet.

## Hvor godt løser applikasjonen oppgaven

Jeg er veldig fornøyd med hvordan applikasjonen løser oppgaven. Av kravene som er listet opp til oppgaven er alle oppfylt i applikasjonen. Selv mener jeg at applikasjonen er enkel å bruke og at det er tilstrekkelig med forklaringer i applikasjonen som gjør det mulig å bruke den helt uten tidligere erfaringer fra andre GIS. Applikasjonen er ikke helt som QGIS enda men det er heller ikke meningen. Den er enkel og har ikke altfor mye funksjonalitet, men jeg mener den har det som er nok og ønskelig i en grunnleggende GIS.

## Brukergrensesnittet

Brukergrensesnittet er bygget opp ganske likt med andre GIS jeg har vært borti. En stor del av applikasjonen er forbeholdt kartet mens en liten andel er satt av til GIS-funksjonalitetene. I min applikasjon har jeg valgt å ha et eget pop-up vindu for GIS-verktøyene. Dette gjør jeg har god plass til å bruke på å gi en oversikt over lagene som blir vist. Kolonnen til venstre, som er den delen av applikasjonen som ikke brukes av kartet består av to deler. Den ene delen er knappene og den andre er området hvor lagene blir synlig. Knappene lar deg laste opp lag, åpne hjelp vinduet og åpne verktøy vinduet. Området over brukes til å vise lagene på liste format. Her er det også mulig å endre synlighet, farge, gjennomsiktighet og rekkefølgen på hvert enkelt lag.

Hjelpevinduet er åpent automatisk når man åpner applikasjonen og kan også åpnes på senere tidspunkt. Her får en generell informasjon om applikasjonen, bruken og det kartografiske verktøyene som kan brukes.

Verktøvinduet åpnes ved å klikke på "Spatial tools" knappen. Da åpnes et vindu hvor en kan velge hvilken operasjon man vil gjøre og på hvilke lag.

Brukergrensesnittet er utformet på en måte som skal gjøre applikasjonen enkel og intuitiv å bruke. Herav er den relativ enkel med lite ekstra og unødvendige funksjonaliteter. Selv om applikasjonen er minimalistisk mener jeg det fungerer fordi den har de essensielle funksjonalitetene.

## Kode

<https://github.com/martinwe001/programming-in-geo>

## Referansar

<https://reactjs.org/>

<https://leafletjs.com/reference-1.7.1.html>

<https://turfjs.org/>

<https://react-bootstrap.github.io/>

<https://casesandberg.github.io/react-color/>

<https://material-ui.com/>

<https://material-ui.com/components/material-icons/>

<https://github.com/atlassian/react-beautiful-dnd>

## Tutorial

<https://martinwe001.github.io/programming-in-geo/>

Du er ny i Trondheim og skal treffe vennen din Ola. Han sa du skulle treffe han ved broen han bor ved, men du har ikke fått telefonnummeret hans så du kan ikke ringe for å spørre

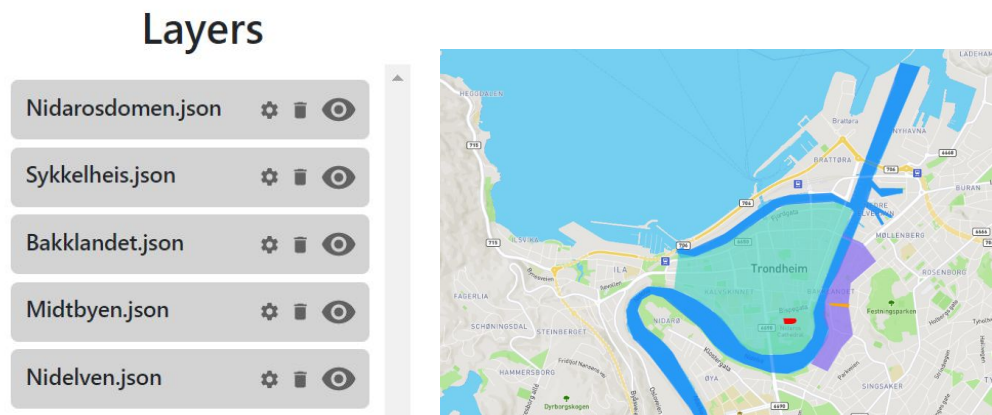
hvor han bor. Heldigvis har han sagt noen ting som gjør at det er mulig for deg ved hjelp av geografisk analyse å finne frem til riktig sted. Dette vet du.

- Ola sier han noen ganger våkner av ringingen til Nidarosdomen, noe han synes var rart siden han ikke bor så nærme at han kan se kirken. Av dette skjønner du at han må bo innenfor 300 m fra kirken, men kan umulig bo nærmere enn 100 m. **2 buffer og difference**
- Ola har også sagt at han bor i Midtbyen, eller var det Bakklandet?. **Intersect**
- Siden Ola bor rett ved en av broene kan du også regne med at han bor maks 100 m fra nidelven.
- Ola har snakket om at han går forbi sykkelheisen ofte, både til og fra skolen, men også når han skal til byen. Du regner med at han bor maks 200 m unna sykkelheisen.

### Bruk av programmet

Du vil få litt informasjon i et hjelpevindu som åpnes automatisk med appen, allikevel vil jeg forklare litt her. Filene finner du her <https://github.com/martinwe001/programming-in-geo> i mappen som heter Layers. Last ned alle filene på din PC.

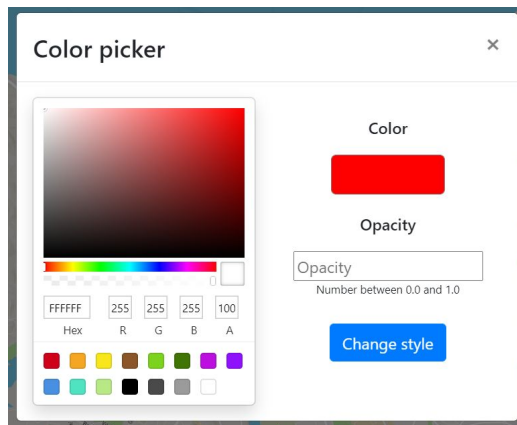
- Filene lastes opp ved å klikke på **Velg fil** - knappen. Her kan du laste opp 1 og 1 fil. Last opp alle filene fra mappen som fulgte med oppgaven.



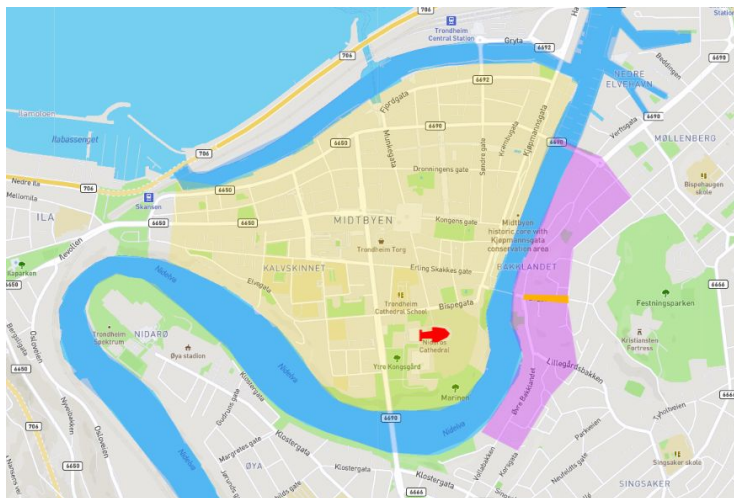
Du burde nå ha dette i vinduet med alle lagene og kartet burde ligne på dette.

- Du kan dra i lagene for å endre på hvilken rekkefølge de vises på kartet. Laget som ligger øverst i listen er den som vises øverst på kartet. *Her kan det være lurt å legge lag som dekker store områder lenger ned og de som ligger oppå andre på toppen.*
- For å endre farge eller gjennomsiktighet, til et lag, klikker du på tannhjulet. Menyen som dukker opp vil se ut som dette.





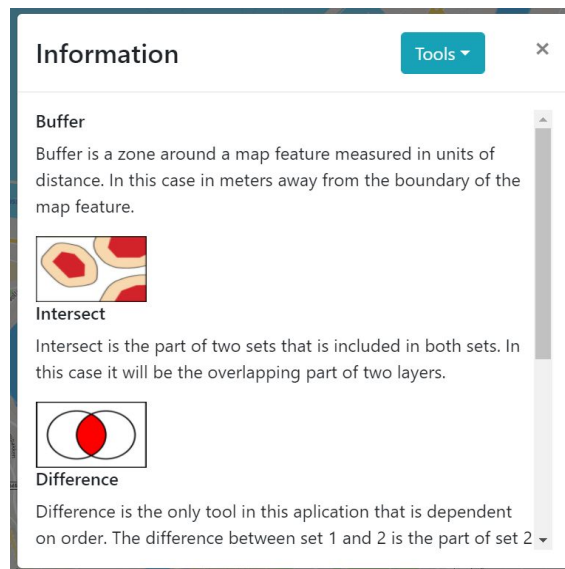
Du kan endre på fargen i vinduet til venstre og fargen du har valgt vil vises under **color** til høyre. I feltet under **color**, **opacity** kan du velge gjennomsiktigheten. Her velger du et tall mellom 0 og 1 hvor 0 er helt gjennomsiktig og 1 er ikke i det hele tatt. *Prøv å endre litt på farger og gjennomsiktighet slik at du får en god oversikt over området og de forskjellige lagene. Etter dette kan det se ut som dette for eksempel.*



- Etterhvert som analysen begynnes kan det være at noen lag ikke er nødvendig å ha der lenger. Da går det an å klikke på øyet for å “skru laget av og på”. Det kan også være relevant å slette laget da klikker du på søppelkassen. *Sletter du et lag du ikke mente kan du bare laste opp eller lage det på nytt.*

### Geografisk analyse

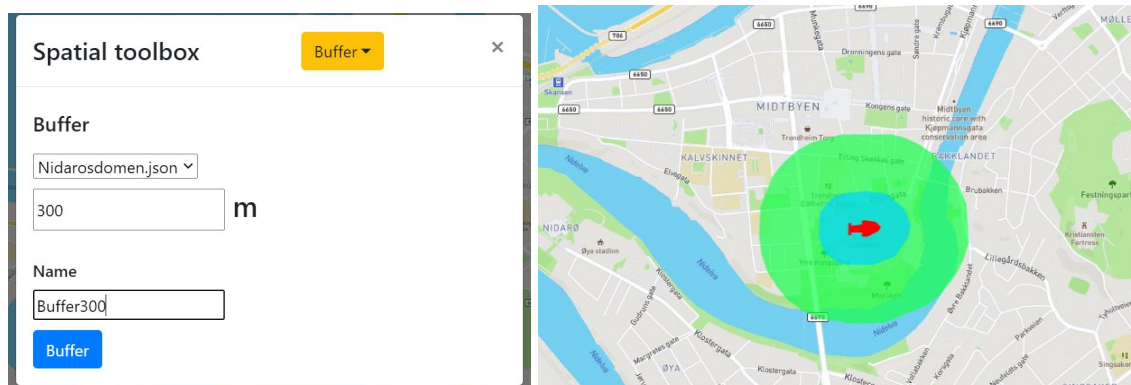
For å gjøre det lettere å forstå hva som gjøres anbefaler jeg å lese seg opp på de forskjellige verktøyene i hjelp menyen. Denne finner du ved å klikke på den gule knappen nederst til venstre. Når vinduet dukker opp klikker du på drop down menyen (blå) og velger **tools**.



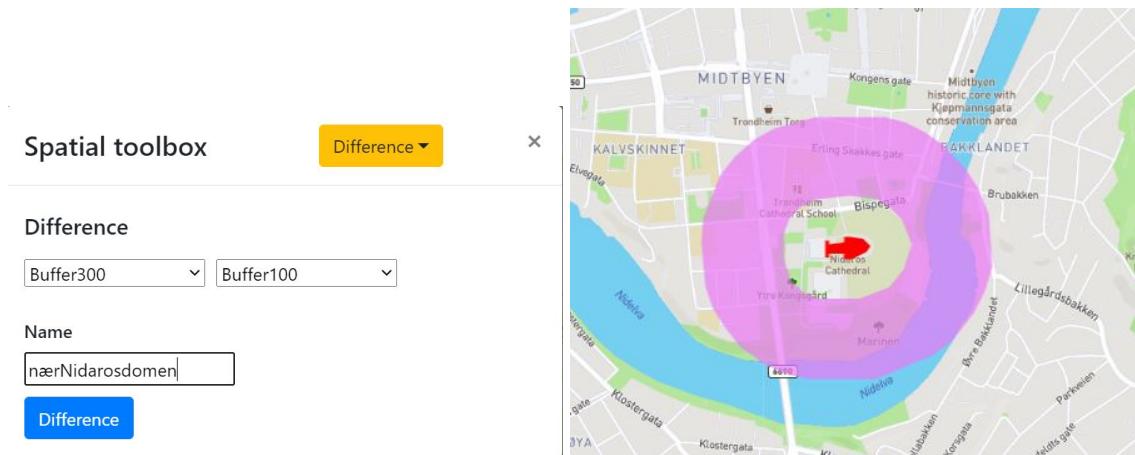
På bildene har jeg valgt å skjule de andre lagene som ikke er i bruk på hvert steg. Dette er for å gjøre det lettere å se hva som gjøres og se om man har gjort det riktig.

For hvert steg må man åpne **Spatial tools** vinduet ved å klikke på knappen. Da kommer det opp et vindu hvor en kan bruke alle verktøyene. For å bytte mellom verktøy benytter du knappen øverst til høyre. På den kan du bytte mellom de 4 forskjellige verktøyene.

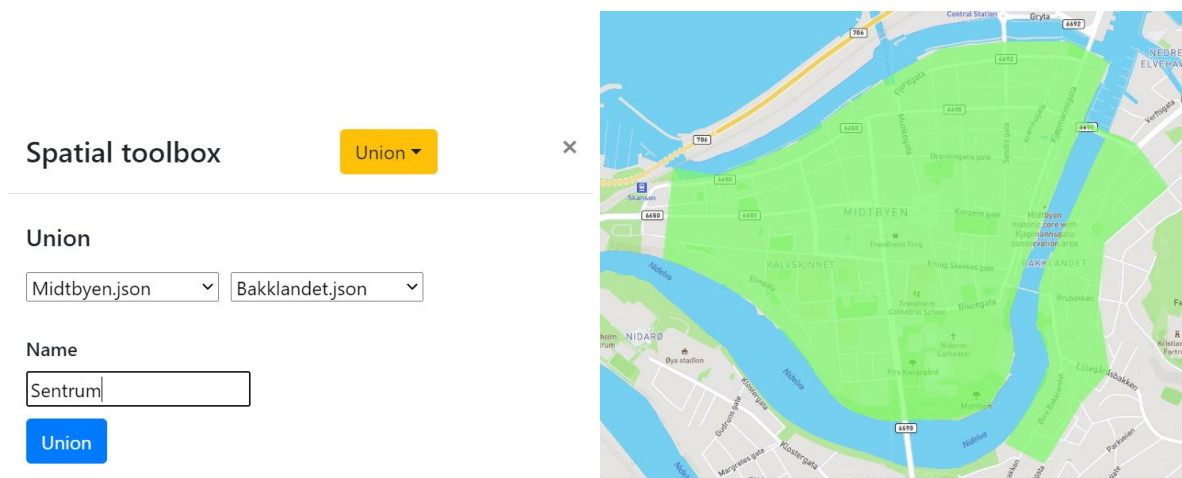
1. **Buffer** rundt **Nidarosdomen.json** på 300m og en til på 100m og kall de henholdsvis **Buffer300** og **Buffer100**.



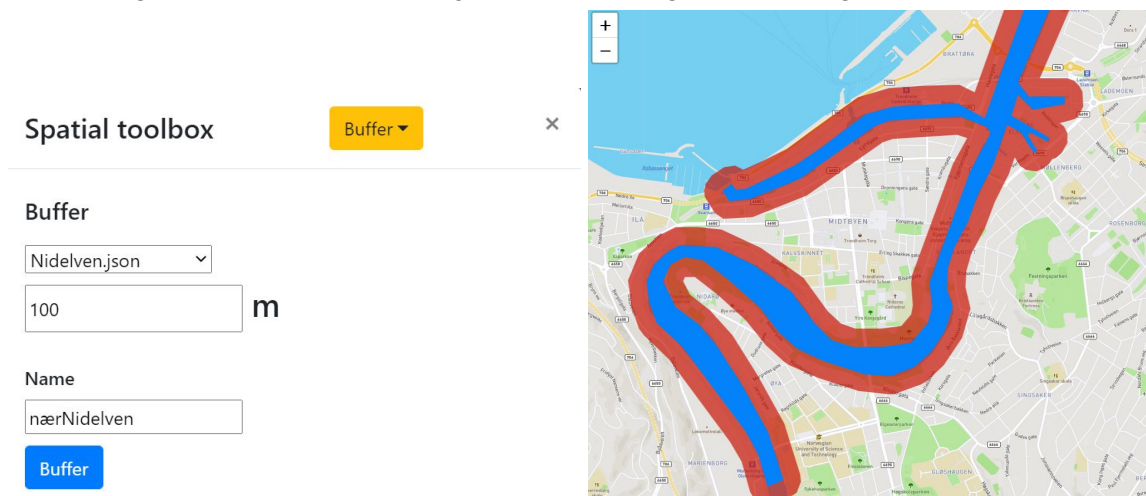
2. **Difference** av **Buffer300** og **Buffer100** og kall dette laget **nærNidarosdomen**. Du kan nå slette **Buffer300** og **Buffer100**.



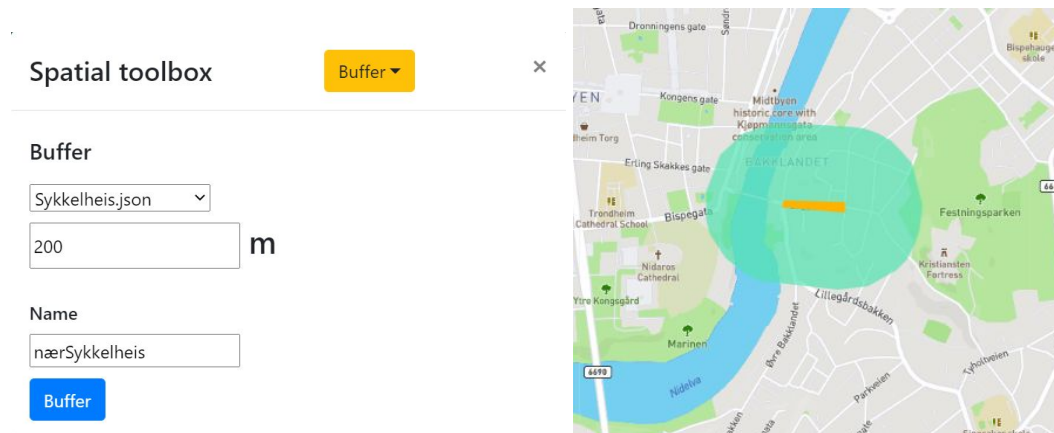
3. Union av **Midtbyen.json** og **Bakklandet.json** og kall dette laget **Sentrum**.



4. Lag **Buffer** rundt **Nidelven.json** på 100m og kall dette laget **nærElven**.

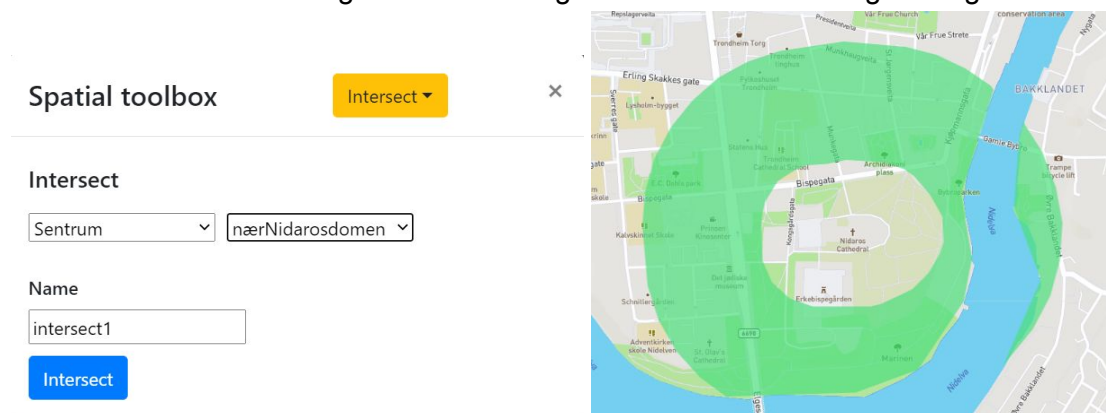


5. Lag **Buffer** rundt **Sykkkelheis.json** på 200m og kall dette laget **nærSykkkelheis**.

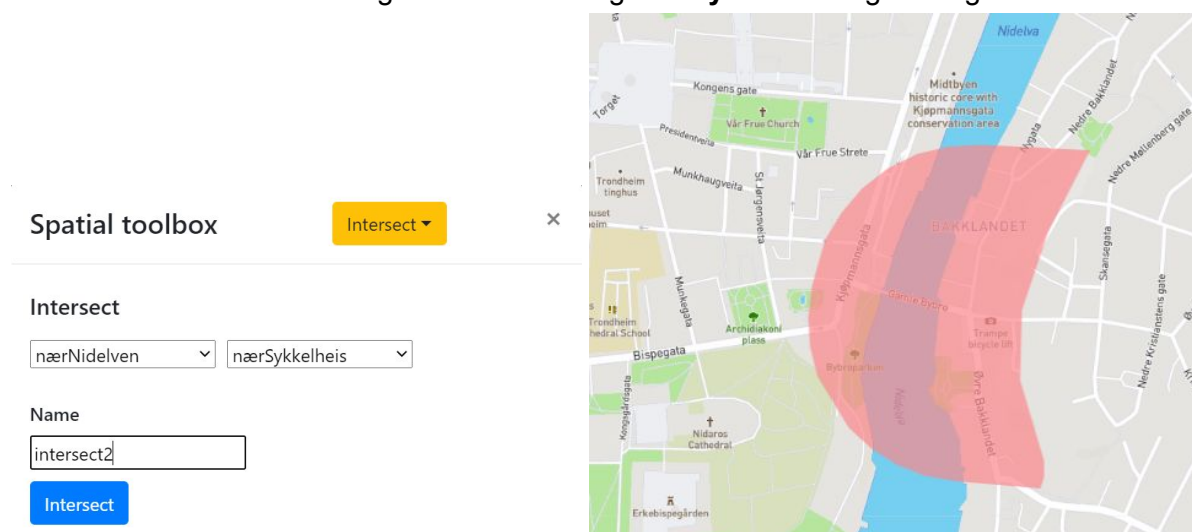


Området Ola bor i må nå befinne seg i krysningsområdet mellom lagene **Sentrum**, **nærNidarosdomen**, **nærElven** og **nærSykkkelheis**. Dette området finner du med **Intersect**.

6. Ta **Intersect** av lagene **Sentrum** og **nærNidarosdomen** og kall laget **intersect1**.



7. Ta så **Intersect** av lagene **nærElven** og **nærSykkkelheis** og kall laget **intersect2**.



8. Ta tilslutt **Intersect** av lagene **intersect1** og **intersect2** og kall det **result**.



Området du nå sitter igjen med er det området Ola må bo i. Ut i fra dette kan du enkelt se hvilken bro du skal treffe Ola på.