

## Visualisering af data

### Introduktion

Denne opgave er stillet som et obligatorisk element af undervisningsforløbet i Unity på datamatiker uddannelsens 4. semester, ved Erhvervsakademiet Dania i Skive.

Formålet med denne opgave er at visualisere data gennem brugen af spilmotoren Unity.

Som en del af besvarelsen på denne opgave, er følgende beskrevet:

- En redegørelse af og baggrunden for dette løsningsforslag.
- En beskrivelse for dataene som ligger til grund for løsningen og hvordan de er anvendt.
- En vurdering af hvad der fremadrettet kan implementeres i løsningsforslaget.

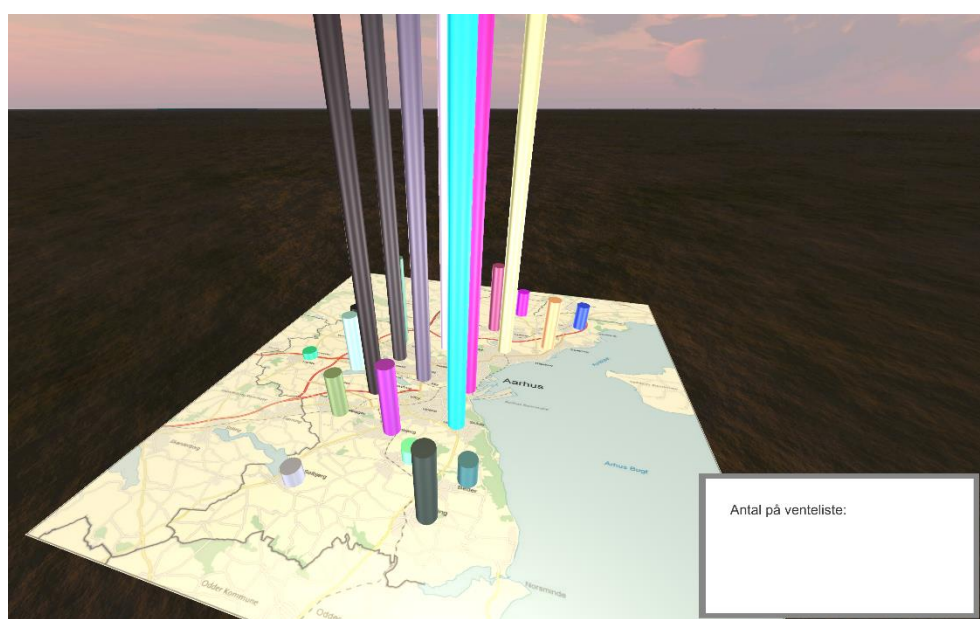
### Løsningsbeskrivelse

Til at løse denne opgave blev der benyttet et datasæt, som kunne beskrive lejeboligmarkedet i Aarhus Kommune. Inspirationen kom da jeg selv har erfaring med lange ventetider på netop lejeboliger i Aarhus Kommune.

Løsningsforslaget giver brugeren en visuel repræsentation af længden af ventelister til lejeboliger i Aarhus Kommune, fordelt på postnumre. Dette gøres ved at spejle en ventelisternes længde i en søjles højde. Højden af de forskellige søjler er derved en direkte repræsentation af hvor mange der står på venteliste til lejeboliger, i det pågældende postområde.

Dertil er de forskellige søjler repræsenteret med farver, for lettere at kunne differencere imellem dem.

Denne repræsentation af ventelisterne og den tydelige forskel i størrelse der forekommer, kan give en bruger, som ønsker en lejebolig i Aarhus Kommune, anledning til at revurdere hvor brugeren ønsker at søge lejebolig.



Det er valgt at implementere et dynamisk kamera, som giver brugeren mulighed for at navigere rundt. Dette blev implementeret da det ellers var svært at danne sig et billede af, hvor stor forskellen egentlig var på søjlernes højde.

Det er også valgt at implementere simple funktioner til programmet. Der er lavet en hjælpe funktion, som beskriver den mulige interaktion med programmet, samt en kort angivelse omkring programmet.

Med den meget store forskel i søjlernes højde, skal antallet af lejeboliger i de forskellige områder også tages for øje. Antallet af lejeboliger må antages at være markant højere i det centrale Aarhus, sammenlignet med det i Beder eller Tranbjerg.

### **Redegørelse for data**

Størstedelen af det data som ligger til grund for løsningen er hentet fra OpenData portalen ([link](#)). Dertil er der hentet mindre elementer fra RegionMidt ([link](#)).

Dataene hentet fra OpenData portalen kom i flere XML-filer, som indeholdte hhv. selskaber, afdelinger, og ventelister. Mens informationen hentet fra RegionMidt, som indeholdte postnumre og områdenavne i Aarhus Kommune, blev lagt direkte i databasen.

XML-filen som indeholdte selskaber var relativt simpel, da den kun indeholde 11 rækker, som hver havde 6 attributter; selskabsID, navn, adresse, postBy, tlf, og email.

XML-filen som indeholdte afdelinger bestod af 444 rækker, som hver havde 7 attributter; afdelingsID, selskabsID, navn, adresse, postBy, husdyr, og hjemmeside.

XML-filen som indeholdte ventelister bestod af 1357 rækker, som hver havde 7 attributter; listeID, afdelingsID, selskabsID, lejemålstype, lejlighedstype, antal rum, og antal på ventelisten.

Ved bearbejdning af XML-filerne, antages det at hvert selskab er et boligudlejningsselskab i Aarhus Kommune, med en adresse på deres kontor. Hver af disse selskaber har flere afdelinger, som udgør en placering for udlejningsboliger. Hver af disse afdelinger kan da have flere lejemål af forskellige typer og størrelser.

Ved import af datene fra XML-filer til MySQL databasen, blev flere attributter sorteret fra. Ved import af afdelinger bevares f.eks. kun selskabsID, afdelingsID, navn, og postnummer. Attributterne der blev sorteret fra vurderes ikke at bidrage til løsningen. Ligeledes ved import af afdelinger blev attributten postby redigeret. Denne attribut indeholdte originalt både en angivelse af postnummer og postby, i samme felt. Da det i løsningen var nødvendigt at kunne sortere rækkerne på deres postnummer, blev angivelsen af postby slettet fra feltet.

Relationen imellem afdelingerne, altså udlejningsboligerne, og ventelisterne er mere kompleks. En afdeling kan have flere ventelister knyttet til sig. Dette skyldes, at en afdeling kan have flere lejemål, f.eks. lejligheder med 2 eller 3 rum. Hver af disse lejlighedstyper eller størrelser har da forskellige ventelister. Der kan derfor være flere ventelister knyttet til de forskellige afdelinger.

Dette bevirker samtidigt at, summeringen af antallet på ventelister på afdelingerne, ikke nødvendigvis er retvisende. Antager man f.eks., at en person ønsker at skrive sig på ventelisten en lejlighed ved afdeling 1, er der umiddelbart intet til hindrer for, at personen både skriver jeg på ventelisten til en 2 værelseslejlighed og til en 3 værelseslejlighed. Dette er ikke direkte forkert, da men kan være interesseret i begge typer af lejligheder, med det giver et misvisende billede af størrelsen af ventelisterne.

### **Fremtidig implementering**

Dataene fra OpenData portalen indeholdte som nævnt informationer omkring selskaber, afdelinger og ventelister. På OpenData portalen er der dog tilsyneladende også muligt at indhente data omkring de specifikke lejemål som er forbundet med de forskellige afdelinger, samt information om til- og fraflytninger fra lejemål. Det var for mig desværre ikke muligt at indhente dette data. Da jeg forsøgte at indhente dataene, gav min browser en timeout fejl, da det tilsyneladende tog for længe at hente dataene.

Havde det været muligt at indhente dette data, kunne løsningsforslaget have set anderledes ud. Med denne data kunne man f.eks. beregne en ventetid på lejeboliger i de forskellige postområder, eller beregne forholdet imellem antallet på venteliste og antallet af lejeboliger.

Med det nuværende data fra OpenData portalen vil det også være muligt at tilføje andre visuelle repræsentationer til løsningsforslaget; f.eks. kan man angive hvilke typer af lejebolig (mht. antal rum, type, osv.) der er størst venteliste på.