

Opgaver uge 7

Tirsdag 11. marts 2008

2008-03-10

Formål med opgaverne

Formålet med disse opgaver er at få begyndt på design og programmering i jeres projekt.

Første skridt er at kunne vise vejdatasættet grafisk, ved at tegne vejene på en Java GUI-komponent (fx `javax.swing.JPanel`). Forelæsning nummer 1 (<http://www.itu.dk/people/sestoft/bachelor/faap-07.pdf>) fra 11. marts viser et eksempel på sådan et skærbillede med alle Sjællands veje, tegnet i forskellige farver afhængig af vejens type. På det skærbillede er motorveje røde og flertallet af andre veje sorte. Områderne med grønt er mest skove, det er fordi vi snedigt har valgt at tegne stier med grønt, og stier findes fortrinsvis i skove og plantager.

Jeres visualisering kan laves på mange forskellige ambitionsniveauer, og kan med fordel udarbejdes gradvis, så I starter med at lave simple faciliteter først og udvider efterhånden. Her er nogle muligheder:

1. Lav et Java-program der kan tegne alle veje fra et givet datasæt eller udtræk (fx Valby, Charlottenlund, Sjælland, hele datasættet), på en GUI-komponent (fx `JPanel`) af en på forhånd givet fast størrelse.
2. Som ovenfor, men vejene tegnes med forskellige farver afhængig af vejtype.
3. Som ovenfor, men nu skal tegningen tilpasse sig `JPanel`'s størrelse når brugeren ændrer den ved at maksimere Java-vinduet eller trække i dets ramme for at gøre det større eller mindre.

Vink: Det er nyttigt at indføre en klasse `Coordinates` som er ansvarlig for omregningen mellem vejdatasættets koordinater og skærmens koordinater, dvs. mellem UTM32-koordinaterne (i meter) i vejdatasættet og `JPanel`'s grafikkoordinater (i pixels). Klassen skal i hvert fald have disse metoder:

- `public int gFromX(double x)` omregner fra UTM32 øst-koordinat `x` i meter til grafikens vandrette koordinat i pixels
- `public int gFromY(double y)` omregner fra UTM32 nord-koordinat `y` i meter til grafikens lodrette koordinat i pixels

Det er hensigtsmæssigt at lave et nyt `Coordinates`-objekt hver gang vinduet har skiftet størrelse. For eksempel kan `JPanel`'s `paint`-metode lave et nyt `Coordinates`-objekt hver gang den bliver kaldt, for den kaldes i hvert fald hver gang vinduet skifter størrelse.

4. Som ovenfor, men brugeren skal kunne zoome ind på vejnettet ved at trække med musen fra hjørne til hjørne i det delrektangel af vinduet der skal fokuseres på.

Vink: Eftersom man skal kunne omregne fra musekoordinater (altså grafikkoordinater) til vejdatakoordinater skal der bruges flere metoder i `Coordinates`-klassen:

- `public double xFromG(int gX)` omregner fra grafikens vandrette koordinat `gX` i pixels til UTM32 øst-koordinat i meter
- `public double yFromG(int gY)` omregner grafikens lodrette koordinat `gY` i pixels til UTM32 nord-koordinat i meter

Vink: Zoom betyder dels at kun nogle af vejstykkerne skal tegnes, og dels at koordinatomregningerne skal ændres lidt.

5. Som ovenfor, men nu med mulighed for at zoome ud igen, fx blot ved at trykke på tasten `o` (for "out").

Vink: Konstruktoren i et `JPanel` skal udføre `this.setFocusable(true)` hvis panelet skal kunne optage tastetryk. Selve reaktionen på tastetryk sker i en `KeyListener` som man knytter til panelet.

6. Som ovenfor, men med mulighed for at scrolle kortet mod øst, syd, vest eller nord, for eksempel ved brug af pilene på tastaturet.

Vink: Scrolling betyder ligesom zoom dels at kun bestemte vejstykker skal tegnes, og dels at koordinatomregningerne skal ændres lidt.

7. Som ovenfor, men når man flytter musen hen over det tegnede kort vises navnet på den nærmeste vej i en statuslinje (uden at man klikker med musen).
8. Der kan tilføjes menuer til at vælge at kun bestemte typer veje (fx motorveje, primærruter, stier) skal vises. Eller kun de veje som er relevante for biler (altså ikke stier), eller kun dem der er relevante for cyklister (altså ikke motorveje). Eller kun bestemte postnumre.
9. Der er talløse andre muligheder for visualisering. Find gerne på flere.

Gruppen bestemmer selv hvor mange visualiseringsfaciliteter den vil implementere i den endelige projektaflevering. **Første delaflevering bør omfatte i hvert fald trinnene 1-4 nævnt ovenfor**, og skal bestå af én zip-fil med følgende indhold:

- En beskrivelse af hvilke visualiseringsfaciliteter der er valgt og programmeret i første omgang, med en kort begrundelse. (Ca. 2 sider).
- En beskrivelse af hvordan visualiseringerne er implementeret, med en tegning af skærbilledet der forklarer koordinatomregningern. (Ca. 4 sider).
Ovenstående skal være indeholdt i ét dokument med en forside der angiver gruppens farvekode og medlemmer, dato, og kontekst for afleveringen (Førsteårsprojekt i BSc softwareudvikling på IT-Universitetet).
- Det Java-program der implementerer visualiseringerne.

Dokumenterne skal afleveres, med én aflevering per gruppe, til både Claus (brabrand@itu.dk) og Peter (ses-toft@itu.dk) **senest tirsdag 1. april kl 1200**, eller på papir. Dokumenterne kan senere indgå i redigeret form i gruppens projektrapport. Programdelene kan indgå uændret eller med alle de ændringer I ønsker.

Tirsdag den 8. april skal hver gruppe desuden holde en kort fremlæggelse og demonstration af den visualisering gruppen har lavet. Fremlæggelserne laves i to blokke, så tre andre grupper lytter på hver gruppe der præsenterer og kan fremsætte konstruktive forslag. Tidsplan og nærmere detaljer kommer på kursushjemmesiden.