

Indholdsfortegnelse

| | | |
|----------|---------------------------------------------------|-----------|
| 1 | Indledning | 3 |
| 1.1 | Introduktion | 3 |
| 1.2 | Omkring Bilag | 3 |
| 2 | Introduktion til Problem | 4 |
| 3 | Problemanalyse | 5 |
| 3.1 | Datastruktur | 5 |
| 3.2 | Translator | 5 |
| 3.3 | Anti-aliasing | 6 |
| 3.4 | Vector og Box | 6 |
| 3.5 | Nærmeste vej | 7 |
| 4 | Implementationsbeskrivelse | 8 |
| 4.1 | Translator | 8 |
| | setLines() | 8 |
| | translateToView() og translateToModel() | 8 |
| 5 | Afprøvning | 9 |
| 5.1 | Afprøvning | 9 |
| | Metode | 9 |
| | Ækvivalensklasser | 9 |
| | Dokumentation | 10 |
| 6 | Evaluering af programmet | 11 |
| 6.1 | Hastighed | 11 |
| 6.2 | Fejl og mangler | 11 |
| 6.3 | Brugervenlighed | 12 |
| 7 | Evaluering og refleksion over proces | 13 |
| 7.1 | Evaluering og refleksion over proces | 13 |
| 8 | Konklusion | 15 |
| | Bilag | 16 |

| | | |
|-----|--------------------------------|----|
| 8.1 | Kravsspecifikation | 16 |
| 8.2 | Brugsvejledning | 34 |
| 8.3 | Krakdokumentation | 34 |
| 8.4 | Arbejdslog | 48 |
| 8.5 | Arbejdsblade | 52 |
| 8.6 | Samarbejdsaftale | 61 |
| 8.7 | Forventningstabeller | 63 |
| | Vector | 63 |
| | Box | 64 |

Indledning

1.1 Introduktion

Denne rapport indeholder gruppe J's analyse og beskrivelse af den implementerede løsning af "Project: part 1: Visualisation". implementationen og rapporten blev udfærdiget i løbet af 4 uger.

1.2 Omkring Bilag

I rapporten har vi vedlagt følgende bilag:

- Kravsspecifikation: Den af kursusansvarlige udleverede PDF
- Brugsvejledning: En simpel vejledning til brug af vores implementering
- Implementeringsdokumentation: Vores Javadoc-genereret dokumentation af vores implementation
- Krakdokumentation: Den medfølgende dokumentation til vores datapakke
- Arbejdslog: Vores log over den ugentlige proces
- Arbejdsblade: Vores log over det udførte arbejde
- Samarbejdsaftale: Vores aftale omkring hvordan vi ville håndtere vores arbejdsforløb
- Forventningstabeller: Oversigt over resultater fra programafprøvnin-gen

Introduktion til Problem

Vores opgave går ud på at lave et interaktivt Danmarkskort, med indbyggede zoom og scroll-funktioner. Kortet skal have alle veje i det udleverede datasæt og stier markeret og være i stand til at vise den nærmeste vej, på der hvor ens cursor er. For den præcise kravspecifikation se "LAV REFERENCE TIL BILAGET MED KRAVSSPECIFIKATIONEN"

Ud over de obligatoriske krav, har vi implementeret følgende features:

- Zoom out: Man kan zoome ud på keyboardet v.h.a. o,+ eller num(+), eller ved at rulle sit mussehjul baglæns
- Zoom in: Man kan zoome ind på keyboardet v.h.a. i,- eller num(-), eller ved at rulle sit mussehjul fremad
- Scrolling: Man kan scrolle kortet på keyboardet via piletasterne eller med musen ved at venstreklikke og hive kortet rundt
- Map reset: Man kan vende tilbage til startposition og zoomgrad på kortet v.h.a. r, backspace eller space, eller ved at trykke på midterste musseknop
- Vi har tegnet Danmarks kystlinje

Problemanalyse

3.1 Datastruktur

I implementeringen af ethvert kort må den bagvedliggende datastruktur nøje overvejes, da denne i høj grad afgør applikationens ressourcekrav. Overvejelserne omkring datastrukturen involverede blandt andet muligheden for kun at indlæse specifikke dele af dataet samt søgeegenskaber. Det givne datasæt består af punkter og tilhørende linjesegmenter, som forbinder to punkter. Punkterne indeholder information om deres individuelle placering, mens linjesegmenterne er forbundet med en række forskellige oplysninger såsom vejnavn og -type. Da kortdataet består af linjesegmenter, er det også af betydning, at disse kan repræsenteres gennemførligt. Værd at overveje var også datastørrelsen - datakilden fra Krak indeholder over en million punkter, og disse skal på ethvert tidspunkt under programkørslen kunne tilgås hurtigt.

To forskellige datastrukturer syntes besidde de nævnte egenskaber: en sorteret tabel og et quadtree. Den sorterede tabel brillere med en ukompliceret implementering. En quadtree-struktur, mens kompliceret at implementere, udmærker sig derimod ved dens simple datasegmentering samt hastige indlæsning af specifikke områder. Derudover er det muligt at justere centrale dele af en quadtree-struktur, så denne indpasses datakilden. Det vurderedes at quadtree-strukturen ville udmunde i en både hurtigere og mere dynamisk datastruktur, og således sås som den mest passende løsning.

3.2 Translator

Vi har valgt at have en central klasse i controller modulet der oversætter data fra modellen til view. Vores argument for den valgte struktur er at vi som følge af MVC tankegangen forsøgte at holde vores model og view adskilt, og derfor havde brug for et mellemlidende der løsnede koblingen mellem model og view.

Alternativt kunne man have modtaget den rå data direkte i view og oversat den der. Dette havde givet væsentligt mindre modularitet, og koblingen mellem model og view havde været fuldstændig.

3.3 Anti-aliasing

Under vores indlende eksperimenter med tegnemetoderne i Swing fandt vi ud af at det var langt mere omkostningsfuldt at tegne kortet med anti-aliasing slået til. I den nuværende implementation tager der i praksis 5 gange længere tid at tegne kortet med anti-aliasing, end det gør at tegne det uden.

Derfor besluttede vi os for at tegne kortet uden anti-aliasing når brugeren interagerede med kortet, med det formål at opnå høje opdateringsfrekvenser. Når brugeren er færdig med at interagere med kortet tegnes det igen efter en kort pause, denne gang med anti-aliasing slået til. Denne idé var blandt andet inspireret af Adobe InDesign der anvender en lignende løsning.

Vi overvejede også at anvende et stilbillede af det tegnede kort som kunne flyttes og skaleres imens brugeren interagerede med kortet, hvilket vi formoder ville give meget høje opdateringsfrekvenser. I stedet valgte vi den nuværende løsning fordi den konstant opdaterer kortet, og samtidig er tilstrækkeligt effektiv.

Havde vi haft behov for at tegne mange flere linjer havde vi muligvis haft brug for en mere effektiv løsning.

3.4 Vector og Box

Til at starte med valgte vi at repræsentere vektorer og bokse (rektangler) som henholdsvis én- og flerdimensionale tabeller (arrays).

Dette virkede umiddelbart som et smart valg, da man på den måde kunne nøjes med at have én variabel per punkt eller boks. Det viste sig dog at være problematisk efterhånden som vi fik skrevet kode der foretog beregninger med vektorer og bokse, da de indeks som værdierne havde i tabellen ikke fortalte noget om hvad værdien indeholdt.

Et andet problem var at vi foretog de samme operationer på tabellerne mange gange i forskellige klasser. Vi indså derfor at vi havde brug for at samle funktionaliteten der havde at gøre med vektorer og bokse for at undgå unødigt kode-duplikering der gjorde kodebasen svær at vedligeholde.

Vores løsning på begge ovenstående problemstillinger blev at implementere to klasser, *Vector* og *Box*, der begge er datastrukturer med tilhørende hjælpemetoder.

Ved at implementere datastrukturer fik vi løst problemet med navngivningen af værdier, der nu blev mere sigende. For eksempel gik vektorens koordinatsæt fra at være værdien tilhørende indeks 0 og 1 i en tabel, til at være værdien tilknyttet *x* og *y* feltet i et *Vector* objekt.

Ved at implementere hjælpefunktioner fik vi løst problemet med kodeduplikering. Derudover muliggjorde det testing af funktionaliteten, hvilket viste sig at være meget brugbart, da der i perioder under udviklingen var mange fejl i den kode der var afhængig af vektorer og bokse som var svære at udbedre. Ved at teste *Vector* og *Box* klassernes hjælpefunktioner havde vi et solidt udgangspunkt, og kunne koncentrere os om de logiske fejl.

Endelig gjorde det arbejdet med datatyperne nemmere at vi designede hjælpemetoderne således at man kunne sammenkæde metoder.

3.5 Nærmeste vej

I kravspecifikationen nævnes muligheden for at se navnet på vejen nærmest markøren som et implementationskrav. Dette måtte formås med en søgealgoritme i data modellen. Udformningen af søgealgoritmen lå imidlertid ikke fast, da flere forskellige løsningsforslag blev fremsat. Datamodellen var implementeret med en quadtree-struktur (se 3.1), hvilket havde afgørende betydning for søgealgoritmens udformning. Vores første indskydelse var at søge efter en knudes nabo i quadtree-struktur, men dette viste sig umiddelbart at være indviklet. Derfor så vi i stedet en løsning som byggedes på allerede eksisterende funktionalitet i modellen - nemlig rektangulære forespørgsler på data. Et løsningsforslag blev således at forespørge om et indledende rektangel med en fastsat størrelse omkring brugerens markør for derefter at udvide dennes dimensioner indtil en eller flere veje findes. Sammenlignet med det førnævnte løsningsforslag med nabosøgning vurderedes denne løsning at være langsommere, men da implementeringen af nabosøgning blev betragtet som værende meget besværlig, valgte vi at arbejde videre med den langsommere løsning af tidshensyn. Det vurderedes endvidere, at hastighedsforskellen mellem løsningerne ville vise sig at være ubetydelig, hvilket styrkede vores valg af søgealgoritme.

Implementationsbeskrivelse

4.1 Translator

Translator er ansvarlig for at oversætte data fra view til model. I en naturlig forlængelse af dette holder den også styr på tilstanden (center og zoom) af det nuværende udsnit af kortet.

setLines()

setLines() henter Edges fra model og konverterer dem til Lines som den sender til view.

Indledningsvis initialiseres en *Box* (*queryBox*) der definerer hvilket udsnit af kortet der skal hentes fra modellen. Dernæst hentes *Edges* fra model, og de konverteres til *Lines* ved hjælp af *translateToView()*. For at undgå at oprette unødvendigt mange *Line* objekter anvendes en simplificeret object-pool (*linePool*).

translateToView() og translateToModel()

translateToView() og *translateToModel()* står henholdsvis for konvertering af *Vectors* fra model til view, og fra view til model.

translateToView() trækker først startvektoren for *queryBox* fra vektoren, og vektoren tager dermed udgangspunkt i øverste venstre hjørne af *queryBox*. Derefter benyttes *translate()* til at skalere vektoren fra *queryBox* til *canvasBox*. Endelig spejlvendes vektorens y-koordinat i forhold til *canvasBox* med *mirrorY()*, hvilket er nødvendigt eftersom koordinatsystemet i model og view tager udgangspunkt i henholdsvis nederste venstre hjørne og øverste venstre hjørne.

translateToModel foretager samme proces, blot i omvendt rækkefølge.

Afprøvning

5.1 Afprøvning

I bestræbelse efter sikring af applikationens kvalitet og pålidelighed er en række afprøvninger af dennes bestanddele udført løbende med udviklingen. Afprøvningen er ikke foretaget med en målsætning om at udtømme samtlige muligheder for systemfejl, men derimod blot for fyldestgørende at forhindre invaliderende defekter samt bidrage til udviklingsprocessen.

Metode

Da afprøvningsmetoden, som anvendes under udarbejdelsen af programprøver, er afgørende for afprøvningens dækningsgrad, fastsattes den indledningsvist som *black-box testing*. Det er således påkrævet at samtlige kode-segmenter i en afprøvet kodeblok køres én gang som minimum.

Eftersom udviklingen er foretaget i Java, er afprøvning udført i samspil med prøvemiljøet JUnit - et populært Java værktøj til netop afprøvning. JUnit er således anvendt under kørsel af prøverne, hvorfor disse også er skrevet op imod JUnits API.

Som nævnt er formålet med afprøvningen ikke at gennemteste samtlige kodefragmenter i applikationen, hvorfor kun en række konkrete klasser og deres metoder er afprøvet. Især *Vector* og *Box* klasserne ses brugt på tværs af applikationen, og derfor er netop disse udvalgt.

Ækvivalensklasser

Vector og *Box* klasserne er dataklasser, som indeholder værktøjsmetoder til hyppige operationer. Derfor håndterer samtlige metoder i bund og grund kommatal. Ækvivalensklasserne for afprøvning af klassernes metoder er således negative input, positive input samt nul-input.

Dokumentation

Dokumentation for de udførte prøver findes i bilag 8.7. Her ses de forventede og faktiske resultater af afprøvningsmetoderne opstillet i forventningstabeller.

Evaluering af programmet

Programmet understøtter alle de påkrævede features, såvel som flere af de valgfrie. Vi er derfor grundlæggende tilfredse med programmets funktionalitet. Vi kunne godt have tænkt os at have været mere kreative, og fundet på nogle features selv, men vi havde ikke det fornødne overskud.

6.1 Hastighed

Programmet er 5 eller færre sekunder om at indlæse data ved opstart, hvilket vi synes er acceptabelt taget i betragtning af at data er lagret i rå tekstfiler.

Vi har arbejdet en hel del med at få applikationen til at have en hurtig opdateringsfrekvens uanset udsnit og type af brugerinteraktion, og det er vores indtryk at det langt hen ad vejen er lykkedes. Dog kan der opstå relativt lave opdateringsfrekvenser når der navigeres i særligt krævende udsnit af kortet (f.eks. helt zoomet ud, eller overblik over København).

6.2 Fejl og mangler

En væsentligt fejl i vores program er at algoritmen der afgør hvilken vej der er tættest på musemarkøren er unøjagtig i langt de fleste tilfælde fordi den ikke beregner afstanden mellem markøren og vejen på en matematisk korrekt måde. I stedet måler den afstanden mellem markøren og vejens centrum, hvilket i mange tilfælde er et rimeligt skøn, men der er intet matematisk belæg for at det er den korrekte vej der bliver fundet. Havde vi haft mere tid ville vi have beregnet afstanden mellem markør og vej korrekt.

Vi er som nævn ovenfor heller ikke helt tilfredse med opdateringsfrekvensen. Vi kan dog ikke umiddelbart komme i tanker om nogen optimering af programmet der vil medføre væsentlige forbedringer uden at ændre programmets funktionalitet. Vores program benytter sig slet ikke af at

"pre-rendrere" kortet, og dette kunne f.eks. implementeres ved at definere et begrænset antal zoom-niveauer.

6.3 Brugervenlighed

Vores program giver ikke brugeren nogen informationer om hvad mulighederne for interaktion er, og det er derfor op til brugeren selv at udforske programmets features. Særligt galt kunne det gå hvis brugeren ikke er bekendt med digitale kort generelt, og derfor ikke har en forhåndsviden om typiske muligheder for interaktion i denne type program.

Evaluering og refleksion over proces

Vi har en idé om at brugere der er bekendte med Google Maps umiddelbart kan finde ud af at bruge hovedparten af programmets features da mulighederne for interaktion ligner dem i Google Maps. Vi har dog ikke fået efterprøvet denne idé, eftersom vi ikke lavet bruger-tests i denne omgang, men overvejer kraftigt at gøre det i forbindelse med næste fase af projektet.

7.1 Evaluering og refleksion over proces

I dette afsnit reflekteres og evalueres der over hele processen fra problemidentifikation til det midlertidige produkt.

I den indledende fase udarbejdede gruppen i fællesskab en samarbejdsaftale, hvor der blev sat prioriteringer, et ambitionsniveau, præferencer m.m. Efter hvert møde skulle der føres log og udfylde arbejdsblade, samt planlægge arbejdsopgaverne for næste gang. Samarbejdsaftalen var med til at afklare hvilke forventninger man har til samarbejdet, og var med til at sætte et fælles mål for projektet.

Der kunne nu tages hul på den næste fase, hvor problemet identificeres og defineres. Her foretog gruppen en grundig gennemgang af de obligatoriske krav, samt hvilke valgfrie krav der kunne tilføjes. Dernæst afgrænsede gruppen sig i første omgang til at tilføje en zoom-ud funktion udover de obligatoriske krav. Det var på daværende tidspunkt svært at tage stilling til, hvilke valgfrie funktioner vi ville tilføje. Det vigtigste for os var at kunne få lavet et første udkast af visualisering af kortet, der får fremvist vejene. På baggrund af dette var gruppen afklaret med at benytte KrakData som datakilde, da den var nemmere at benytte og håndtere.

Det næste trin i forløbet var at få opstillet en løsningsmodel, hvor gruppen tog udgangspunkt i at strukturere koden i et Model View Controller framework. Vi foretog en tavlediskussionen der omhandlede hvilke klasser

der skulle være i hver task. Dette var med til at give gruppen et godt overblik over hvad programmet skulle indeholde, hvilke arbejdsopgaver der skulle udføres og hvilke centrale klasser der skulle laves.

Efter at have opstillet en løsningsmodel, kunne vi arbejde os hen imod at få lavet et første udkast af programmet. Det var ikke alle der fik skrevet kode i første omgang, det var primært dem, som havde styr på at kode, der tog styringen. Dette kunne ikke undgås, da der i starten var få og store arbejdsopgaver, og det var vigtigt for gruppen at opnå resultater på kortest tid. Grundet de få arbejdsopgaver opstod der tilfælde, hvor kun én person fik skrevet kode, mens de andre ville følge med og komme med input. Dette medførte at produktiviteten var lav, og når der opstod fejl i koden gik hele udviklingsprocessen i stå. Det var også denne del der var tungest og tog længst tid.

Produktiviteten begyndte først at blive relativt høj efter at vi fik tegnet kortet for første gang. Der blev arbejdet på forskellige moduler på en gang og Model View Controller frameworket gav stor glæde og viste sig her at være særlig nyttigt. Der blev i gruppen arbejdet mere effektivt, og der blev tilføjet flere valgfrie funktioner. Vi fik i gruppen mere overskud og større overblik og kunne lave væsentlige ændringer med henblik på at nå det midlertidige produkt.

For at undgå den lave produktivitet, kunne vi have lagt en bedre strategi. Vi kunne have fået dannet os et større overblik og overvejet om vi kunne nedbryde nogle klasser til flere klasser og evt. lave et interface. Vi har i gruppen ikke haft den store erfaring med at lave større programmer, og det bl.a. en af grundene til vi støtte ind på denne problematik. Gruppearbejdet har ellers fungeret godt og samarbejdsaftalen er tildels blevet overholdt. Der har været god kommunikation i gruppen og arbejdsmiljøet har været fornuftigt.

Konklusion

Bilag

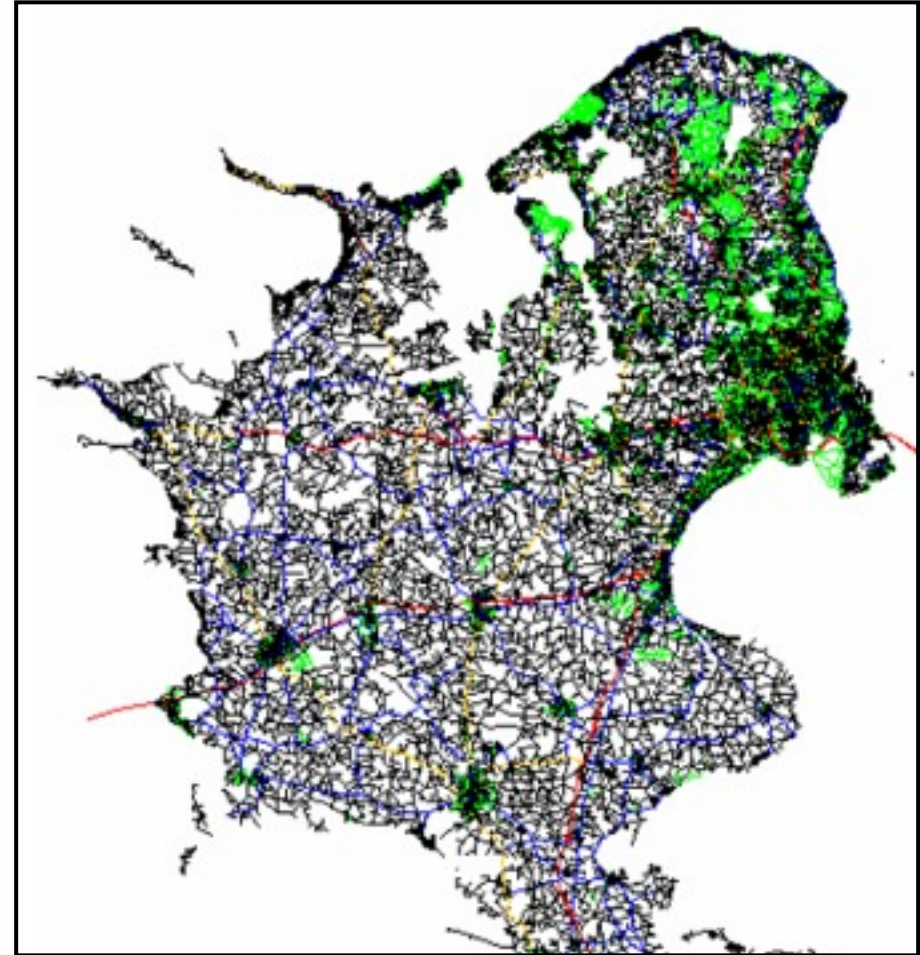
8.1 Kravsspecifikation

Project Part I: Visualisation

Required 1

Draw the map

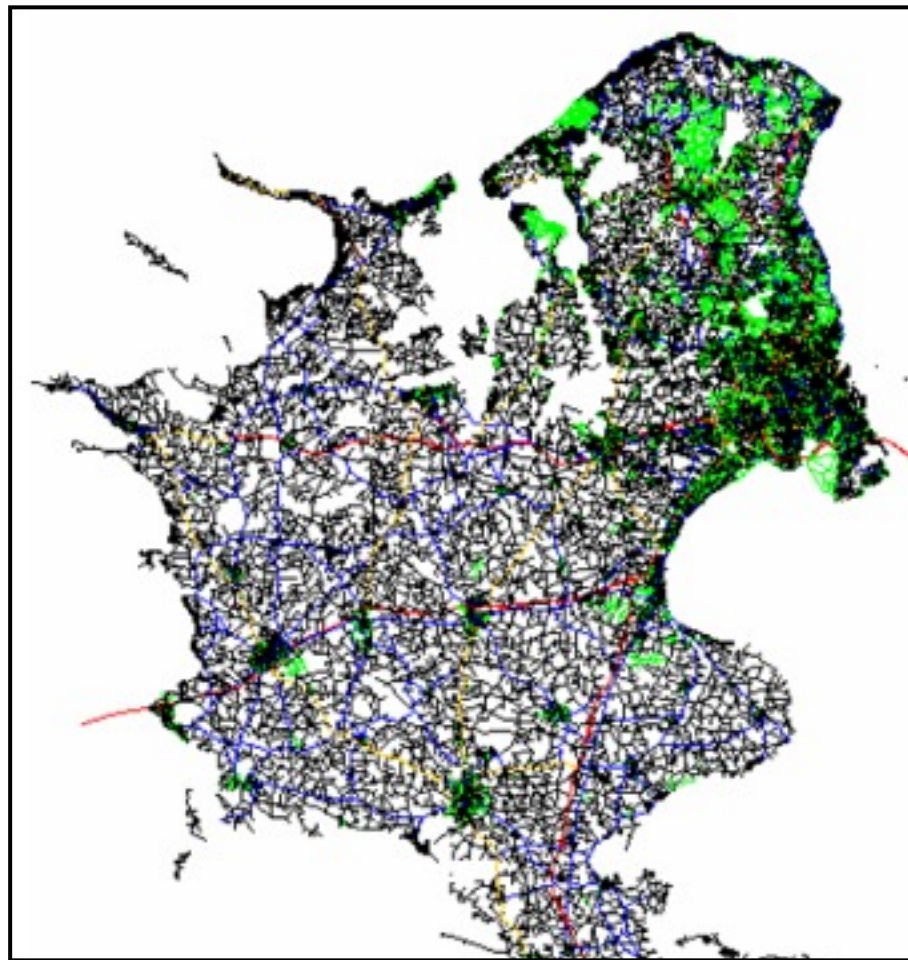
- Make an application that draws all the road segments in the dataset



Required 2

Differentiate between roads

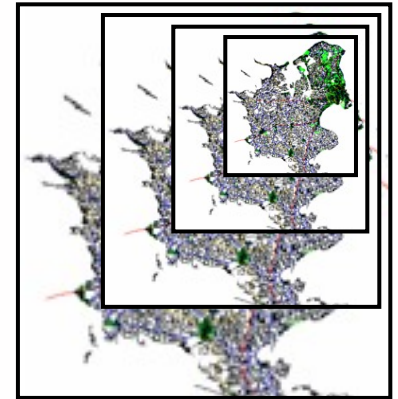
- Draw different segment types with different colours, e.g.
 - Highways: **red**
 - Main roads: **blue**
 - Paths: **green**
 - Other: **black**



Required 3

Adjust size

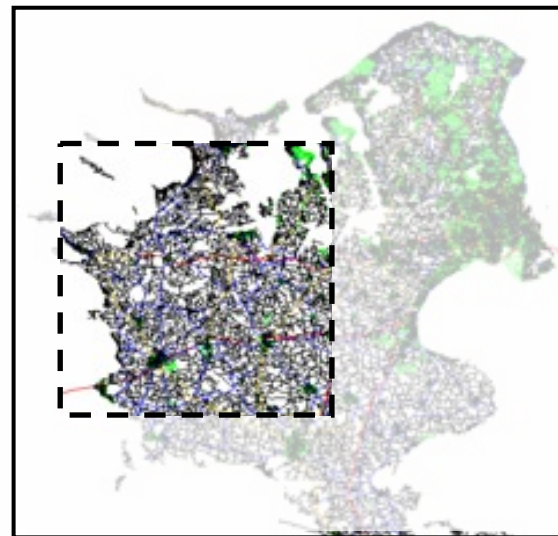
It must be possible to adjust the size of the map drawing, i.e. when resizing the panel with the mouse or maximizing the window.



Required 4

Zoom in

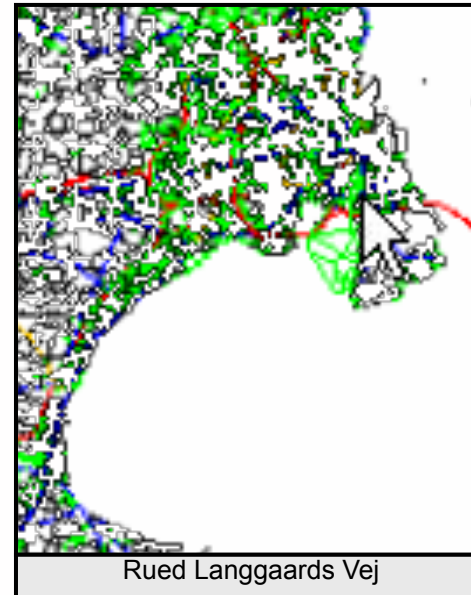
- You must be able to focus on a specific area of the map by dragging a rectangle with the mouse.
- Zoom means that not all road segments are shown and that pixel coordinates change



Required 5

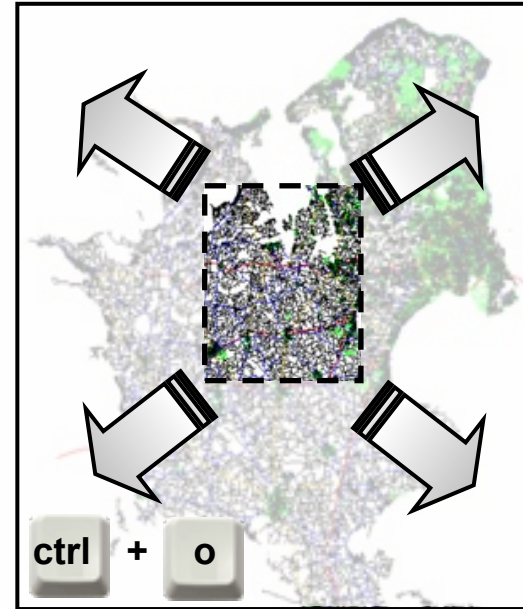
Display closest road name

- Either continuously or when a user **hovers** over the map, unobtrusively display the name of the road closest to the mouse, e.g., in a status bar.



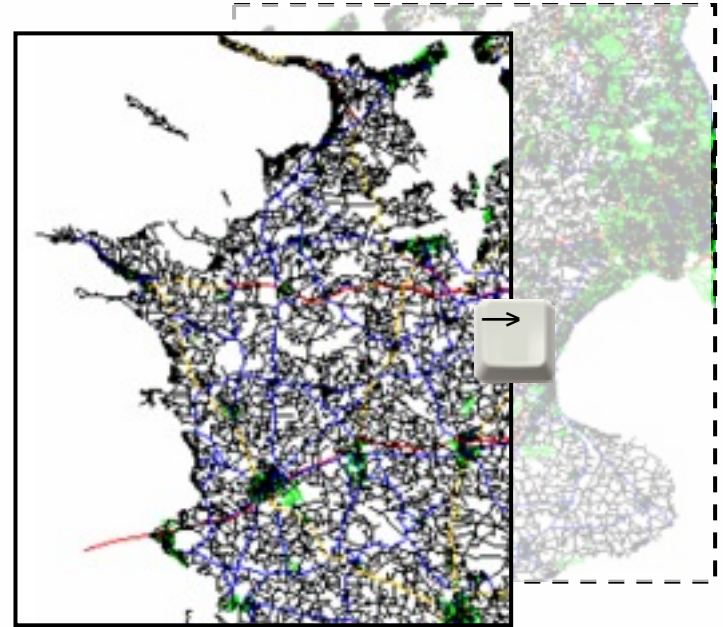
Zoom out

- User must be able to zoom out e.g. by using the mouse wheel or by pressing the key 'O' or Ctrl+O etc.



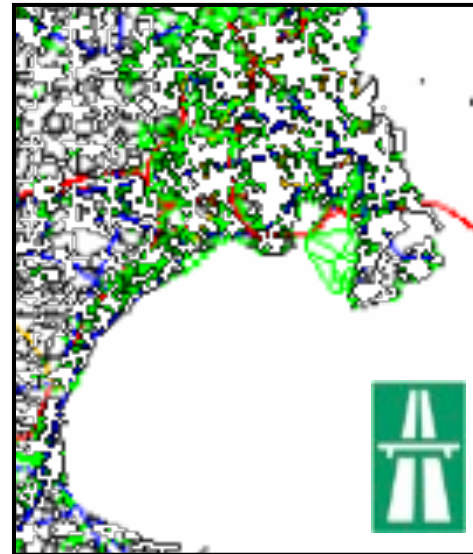
Scrolling

- You should be able to scroll the map by using e.g. the arrow keys or the mouse



Display certain types of roads

- Add options so that user can decide to display only **certain kinds of roads**
e.g. Motorways, paths etc.
- Alternatively this can be that the user chooses transport medium and thus e.g.
 - **No paths** for cars
 - **No motorways** for bicycles



Draw coastline

- A coastline file from OpenStreetMaps, also containing regional boundaries, can be made available to you by the course manager or you can create your own at e.g. here http://www.ngdc.noaa.gov/mgg_coastline/
- (lon/lat format!)

Be creative! Add you own features! Make it better than Google Maps

Possible datasets:

1. Use Krak ASCII dataset (has accompanying Java loader)
2. Use OpenStreetMaps data (data only, no program sources)

Language:

You are encouraged, but not required to use Java.

Important dates

- **Wed. Apr. 2 2014 @ 2pm**
 - **Deadline for the source code and the reports**
- **Mon. Apr. 7 2014 10-16**
 - groups will present a 15' demo of their products
+5mins for questions

What to deliver

At the deadline

- Source code: .java files not .class files
- A report:
 - .pdf up to 12 pages
 - group constitution

Every Monday (apart from April 1):

- worksheets
- group diaries

ALL SUBMISSIONS IN LEARNIT!

One submission pr. group.

- **Use step-wise refinement:**
 - First analyze, design and implement a simple solution then make a more complicated solution
- **Define milestones e.g. 1-2 every week:**
 - Work on a specific required part of the project and then after milestones have been reached ,work with the extensions
- **Be ambitious**
 - You are lot of people with lots of time.
- **Remember Testing!**
 - Not required to hand in now but will be part of the final project

Questions, face to face meeting etc.

Office hours.

- **See learnit (calendar, top-left).**

Have fun!

8.2 Brugsvejledning

Programmet kan køres via den udleverede .jar-fil. Efter start af programmet kan det tage op til 5 sekunder før kortet ses på skærmen.

Tabel 8.1 – Tastaturgenveje

| Kommando | Tast | Tast | Tast |
|------------------------------------------------------|--------------|-------|-----------|
| Zoom kortet ind | i | + | num+ |
| Zoom kortet ud | o | - | num- |
| Returner kortet til startzoom-grad og centrér kortet | r | space | backspace |
| Bevæg kortet | piletasterne | | |

Tabel 8.2 – Mussemanipulation

| Kommando | bruger |
|------------------------------------------------------|--------------------------------------------------------|
| Zoom kortet ind | Rul mussehjulet væk fra bruger |
| Zoom kortet ud | Rul mussehjulet hen mod bruger |
| Zoom kortet ind i markeret område | Højreklik, træk og slip så du har det ønskede udsnit |
| Returner kortet til startzoom-grad og centrér kortet | Tryk på den midterste musseknop |
| Bevæg kortet | Venstreklik, træk og slip så du har det ønskede udsnit |

8.3 Krakdokumentation

Introduction to the KRAK roadmap data and associated code

Access to the Krak data

Please note that we have been given access to the Krak roadmap data strictly for teaching and research. The data must not be redistributed to anyone not participating in this course, and should be deleted after the course ends.

Structure of the Krak roadmap data

The most interesting part of the roadmap data are found in the Txt subfolder in the files 'kdv_node_unload.txt' (Node file from here on) and 'kdv_unload.txt' (edge file from here on) which contains the roadmap data exported to a simple text format.

The node file contains a list of nodes representing intersections in the road network. Each node is associated the following data:

Its geographic position is listed as X-COORD and Y-COORD. It is also associated to identifiers KDV# and KDV-ID. We will use KDV-ID as a global identifier, that is it will remain the same if we use a subset of the nodes, while we will keep KDV# continuous from 1 to the number of nodes in the file. The entry 'ARC#' will not be used.

The Edge file contains a set of edges corresponding to road segments between pairs of intersections.

The FNODE# and TNODE# gives the KDV# of the endpoints of the segment. DAV_DK# is used as a continuous identifier similar to KDV# while DAV_DK-ID is used as a global edge identifier remaining the same even if we only use a subset of the edges. Another field that is very important is LENGTH which gives the length of the road segment.

The table below gives interpretations of the remaining fields.

| Field name | Description |
|------------------------|---------------------------------------------------------|
| TYP | Road type indicator (details only available in Danish) |
| VEJNAVN | Road name |
| FROMLEFT/RIGHT | The beginning of the house numbers on the left/right |
| TOLEFT/RIGHT | The end of the house numbers on the left/right |
| FROMLEFT/RIGHT_BOGSTAV | Same for letters associated to houses |
| TOLEFT/RIGHT_BOGSTAV | Same for letters associated to numbers |
| V_SOGNENR | Parish number on the left side of the road segment |
| H_SOGNENR | Parish number on the right side of the road segment |
| V_POSTNR | Left zip code |
| H_POSTNR | Right zip code |
| KOMMUNENR | Not relevant |

| | |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| VEJKODE | Road ID specific for commune (Not relevant) |
| SUBNET | Not relevant for this project |
| RUTENR | Not relevant for this project |
| FRAKOERSEL | Highway turn-off |
| ZONE | Type of area: 10 – City 20 – Holiday housing 30 – Industrial |
| SPEED | ‘Ideal speed’ (treat as speed limit) |
| DRIVETIME | Estimated driving time |
| ONE_WAY | tf = one way in the direction To-From ft = one way in the direction From-To n = No driving allowed <blank> = No restrictions |
| F_TURN | Related to turn restrictions, if you wish to use such information, talk to a supervisor or examine the Danish Krak documentation and /Txt/turn.txt |
| T_TURN | See above |
| VEJNR | KRAK internal road ID (Not relevant for this project) |
| AENDR_DATO | Change date (Not relevant for this project) |
| TJEK_ID | Road segment ID (Not relevant for this project) |

The code framework

The code framework contains detailed comments, which you should read. The following is just a bird-eye view of the code explaining the overall layout.

Basic Graph Library

The basic graph library is contained in the Java package graphlib, and consists of Edge.java, Node.java and Graph.java. It is a simple implementation of an adjacency list representation of a graph.

KrakLoader

The KrakLoader can read and parse the text files supplied by krak and recreate the corresponding graph using the basic graph library.

InstanceCreator

The instance creator can be used to systematically create a small test road network. It loads the krak text files and allows custom filters to include or exclude nodes and edges. The result can be written to new text files. Please read the comments for the instance creator in detail to be able to trace road segments between the original and the new road network.



Kraks Danske Vejnet

Professionel - Rute

Dokumentation

Kraks Forlag AS
Kraks Kort & GIS
Virumgårdsvej 21
2830 Søborg

Telefon: 45 95 65 00
Fax: 45 95 65 65
E-post: geoinfo@krak.dk



Indholdsfortegnelse

| | |
|-------------------------------------|---|
| 1. Installationsvejledning | 1 |
| 2. Adresse- og Vejnetsdatabase..... | 2 |
| 3. Danmarkskort | 7 |



1. Installationsvejledning

Nye brugere af Kraks Danske Vejnet skal kopiere data fra cd-rom til harddisken. Mappen **Krak** kopieres fra cd-rom til c:

Brugere som modtager opdatering, skal slette det gamle Kraks Danske Vejnet, hvorefter den nye kopieres over på harddisken. **HUSK** at flytte alle egne projekter og data, der ligger i mappen **krak**, før du sletter.

Når vigtige filer er blevet gemt andetsteds så kopieres mappen **krak\kdvpro** fra cd-rom til **c:\krak** på din harddisk. (Man kan godt arbejde med dataene via cd-rom drevet, men computeren arbejder hurtigere, når de er gemt på harddisken).

Derefter kan projektet åbnes og Kraks Danske Vejnet er klar til brug.

Adresse-geokodning

Før adresse-geokodning kan foretages skal man:

1. Kopiere alle filer (undtagen readme filerne) fra mappen "dansk_adrstil" under mappen "kdv" til mappen "Esri/av_gis30/arcview/geocode".
2. Derefter oprettes et geokodning indeks ved at åbne ArcView, åbne et project og vælg kdv temaet. Med kdv temaet valgt, gå du i properties i menulinen Theme og vælger geokodning. Her vælges 'Danske veje med postnr' som Address Style. Opdateringen tager ca. 3-5 minutter og derefter er du klar til at geokode dine adresser. **Husk at gemme projektet og lukke ArcView, så dit nyoprettede geokodningsindeks gemmes.**

Bemærk: Man skal slette alle skivebeskyttelses formater på de filer man vil arbejde med. Dvs. at man skal fjerne skrivebeskyttelsen fra alle filer i vejdatatabasen ved at fremhæve dem (klikke på dem), vælg *egenskaber* efter højre museklik eller fra menuen *filer*, og så fjerne fluebenet i feltet *skrivebeskyttet*. Dette kan klares nemt fra stifinderen.

2. Adresse- og Vejnetsdatabase

Mappen **KDV** indeholder hele det danske vejnet inkl. adresser med husnummerintervaller på de enkelte vejstykker.

Ophavsrettigheder: Kraks Forlag AS har rettigheder til viderebearbejdning, distribution og salg af databasen. DAV (Dansk Adresse- og Vejdatabase) har ophavsret til den grundlæggende database.

Dækningsområde: Hele Danmark.

Nøjagtighed: Vejsegmenterne har en geometrisk nøjagtighed på ca. 5m udtrykt som punktmiddelfejl.

Definition: Mellem to veikryds vil der være mindst et vejsegment. En ændring af en eller flere attributværdier (f.eks. postnummerændring mellem to kryds) vil bevirke, at der vil være mere end et vejsegment mellem de to kryds. Dvs. at vejen af tekniske grunde kan blive delt op i mindre segmenter uden at det er umiddelbart synligt på kortet. Vejsegmenterne udgør et samlet netværk, som gør det muligt at foretage afstandsberegning, mm.

Koordinatsystem: System 34 - Sjælland/Jylland, System 45 - Bornholm - eller aftalt utm-projektion

Kortenheder: Meter

Format: ArcView GIS Shapefil format

Bemærk: Som følge af en uens politik omkring husnummer tildeling i de enkelte kommuner, følger husnumrene ikke altid datamodellen. Husnumrene kan også være ulige på begge sider af vejen, stige på den ene side og falde på den anden side etc.

I mappen **KDV** findes det danske vejnet som følgende ArcView-shapefiler:

| Kdv.shp – Vejnet for hele Danmark | |
|---------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Tabelattributter:</i> | <i>Forklaring:</i> |
| Shape | Line (linie) |
| Length | Længde på vejsegment |
| Vejnavn | Vejnavn og færgerutenavn |
| Typ - indeholder alle <i>type</i> definitioner der ligger i databasen. Typ identer med kursiv skrift er ikke taget i brug fra 1.1.02. | 1 – Motorveje 2 – Motortrafikvej 3 – Primærrute > 6 meter 4 – Sekundærrute > 6 meter 5 – Vej 3 - 6 meter 6 – Anden vej 8 – sti 10 – markvej 11 – gågader 21 – proj. motorvej 22 – proj. motortrafikvej 23 – proj. primærvej 24 – proj. sekundærvej 25 – Proj. vej 3-6 m 26 – Proj. vej < 3 m 28 – Proj. sti 31 – Motorvejsafkørsel 32 – Motortrafikvejsafkørsel 33 – Primærvejsafkørsel 34 – Sekundærvejsafkørsel 35 – Anden vejafkørsel 41 – Motorvejstunnel 42 – Motortrafikvejstunnel 43 – <i>Primærvejstunnel</i> 44 – <i>Sekundærvejstunnel</i> 45 – <i>Anden vejstunnel</i> 46 – <i>Mindre vejstunnel</i> 48 – <i>Situnnel</i> |

| Kdv.shp – Vejnet for hele Danmark | |
|------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Tabelattributter:</i> | <i>Forklaring:</i> |
| | 80 – Færgeforbindelse 99 – Stednavn eksakt beliggenhed ukendt (f.eks. et torv hvor der ingen vejadresse fandtes i det originale dataset.) |
| Fromleft | Husnummer start på segmentets venstre side |
| Toleft | Husnummer slut på segmentets. venstre side |
| Fromright | Husnummer start på segmentets højre side |
| Toright | Husnummer slut på segmentets højre side |
| Fromleft_b | Laveste husbogstav på segmentets venstre side |
| Toleft_bog | Højeste husbogstav på segmentets venstre side |
| Fromright_ | Laveste husbogstav på segmentets højre side |
| Toright_bo | Højeste husbogstav på segmentets højre side |
| V_sognenr | Sognekode på segmentets venstre side |
| H_sognenr | Sognekode på segmentets højre side |
| V_Postnr | Postnummer på segmentets venstre side |
| H_Postnr | Postnummer på segmentets højre side |
| Kommunenr | Unikt kommunenummer |
| Vejkode | Unik vejkode indenfor hver kommune |
| Rutenr | Unikt rutenummer |
| Frakoersel | Motorvejsfrakørsel |
| Zone | 10 - By 20 - Sommerhus 30 - Industri Værdierne er vejledende, idet der forestår en opdatering før de er synkrone med de respektive områder i "Danmarkskort"-mappen. |
| Speed | Idealhastighed på vejsegment |
| Drivetime | Kørselstid pr. segment ved idealhastighed + 15% |
| One_way | tf = ensrettet modsat digitaliseringsretning (To-From) ft = ensrettet i digitaliseringsretning (From-To) n = ingen kørsel tilladt (henholdsvis typ 8, 21-28) <blank> = ingen ensretning |
| Aendr_dato | Dato for eventuelle ændringer i grunddata. |
| Vejnr | Kraks interne vejnummer som kan bruges med andre Kraks produkter. |
| Tjek_id | Unikt identifikationsnummer for alle vejsegmenter |

| Kdv.shp – Vejnet for hele Danmark | |
|------------------------------------------|--------------------|
| <i>Tabelattributter:</i> | <i>Forklaring:</i> |
| | |

I mappen **KDV** forefindes foruden vejnettet også en fil indeholdende informationer om svingrestriktioner. Filen er en negativ liste indeholdende information om hvilke vejsegmenter der IKKE må køres fra/til (F_tjek_id/T_tjek_id). Desuden indeholder turn.shp information om retningen af F- og T-segmentets retning i forhold til den retning der køres mellem de to segmenter - denne information er nødvendig for at tage hensyn til beskrivelsen af u-sving.

Ved et forbud mod u-sving er F- og T-tjek_id ens - derfor anvendes F_edge_dir hhv. T_edge_dir til at beskrive i hvilken ende af segmentet forbudet gælder. Således vil F_edge_dir og T_edge_dir indeholde FT hhv. TF såfremt restriktionen gælder i T-enden af segmentet.

| turn.shp – Svingrestriktioner | |
|--------------------------------------|----------------------------------------------------------|
| <i>Tabelattributter:</i> | <i>Forklaring:</i> |
| Shape | Line (linie) |
| F_tjek_id | Unik ident (tjek_id) på fra vejsegmentet i KDV.shp |
| T_tjek_id | Unik ident (tjek_id) på til vejsegmentet i KDV.shp |
| F_edge_dir | tf/ft - retning af fra vejsegmentet (nødv. pga. u-sving) |
| T_edge_dir | tf/ft - retning af til vejsegmentet (nødv. pga. u-sving) |

I mappen **Vejklasser** er vejene delt op i fire klasser som hver har sit eget tema. Derved kan man nemt lave kort med kun de større veje når små veje ikke er interessante.

| Motorvej.shp – Motorveje | |
|---------------------------------|--------------------------------------|
| <i>Tabelattributter:</i> | <i>Forklaring:</i> |
| Shape | Line (linie) |
| Typ | Udtræk af KDV hvor typ = 1, 31 og 41 |
| Vejnavn | Tekstbeskrivelse |
| Tjek_id | Unik ident |

| Motortrf_primvej.shp – Primærveje | |
|------------------------------------------|--------------------------------------------|
| <i>Tabelattributter:</i> | <i>Forklaring:</i> |
| Shape | Line (linie) |
| Typ | Udtræk af KDV typ = 2, 3, 32, 33, 42 og 43 |
| Vejnavn | Tekstbeskrivelse |
| Tjek_id | Unik ident |

| Sekund_vej.shp – Sekundærveje | |
|--------------------------------------|--------------------------------------|
| <i>Tabelattributter:</i> | <i>Forklaring:</i> |
| Shape | Line (linie) |
| Typ | Udtræk af KDV hvor typ = 4, 34 og 44 |
| Vejnavn | Tekstbeskrivelse |
| Tjek_id | Unik ident |

| Andre_veje.shp – Andre veje | |
|------------------------------------|---------------------------------------------------|
| <i>Tabelattributter:</i> | <i>Forklaring:</i> |
| Shape | Line (linie) |
| Typ | Udtræk af KDV hvor typ = 5,6,11,10,13,35,45 og 46 |
| Vejnavn | Tekstbeskrivelse |
| Tjek_id | Unik ident |

Bemærk: Kraks Forlag garanterer ikke at *alle* attributfelter er udfyldt. Dette gælder f.eks. ensretninger og rutenumre.

3. Danmarkskort

Mappen **Danmarkskort** på cd'en i Kraks Geoinfo Pakke indeholder kortdata, som gør det muligt at fremstille kartografiske kort over hele Danmark.

Ophavsrettigheder: Kraks Forlag AS
Format: ArcView GIS Shape-filformat

I mappen findes følgende af Danmarkskortets elementer som ArcView-shapefiler:

| Danmark.shp - Danmarkskort | |
|-----------------------------------|---------------------------------------|
| <i>Tabelattributter:</i> | <i>Forklaring:</i> |
| Shape | Polygon (flade) |
| Area | Områdeareal (m2) |
| Perimeter | Områdeomkreds (m) |
| Type | 6 – Land 7 – Vand 10 - Tyskland |

| By.shp – byer/bebyggede områder | |
|----------------------------------------|--------------------------|
| <i>Tabelattributter:</i> | <i>Forklaring:</i> |
| Shape | Polygon (flade) |
| Area | Områdeareal |
| Perimeter | Områdeomkreds |
| Navn | Områdets eventuelle navn |

| Industri.shp – industriområder | |
|---------------------------------------|--------------------------|
| <i>Tabelattributter:</i> | <i>Forklaring:</i> |
| Shape | Polygon (flade) |
| Area | Områdeareal |
| Perimeter | Områdeomkreds |
| Navn | Områdets eventuelle navn |

| Sommerhus.shp –sommerhusområder | |
|----------------------------------------|--------------------------|
| <i>Tabelattributter:</i> | <i>Forklaring:</i> |
| Shape | Polygon (flade) |
| Area | Områdeareal |
| Perimeter | Områdeomkreds |
| Navn | Områdets eventuelle navn |

| Soe.shp – natur: søer | |
|------------------------------|--------------------------|
| <i>Tabelattributter:</i> | <i>Forklaring:</i> |
| Shape | Polygon (flade) |
| Area | Områdeareal (m2) |
| Perimeter | Områdeomkreds (m) |
| Navn | Områdets eventuelle navn |

| Skov.shp – natur: skovområder | |
|--------------------------------------|--------------------------|
| <i>Tabelattributter:</i> | <i>Forklaring:</i> |
| Shape | Polygon (flade) |
| Area | Områdeareal (m2) |
| Perimeter | Områdeomkreds (m) |
| Navn | Områdets eventuelle navn |

| Lysning.shp – natur: lysninger | |
|---------------------------------------|--------------------------|
| <i>Tabelattributter:</i> | <i>Forklaring:</i> |
| Shape | Polygon (flade) |
| Area | Områdeareal (m2) |
| Perimeter | Områdeomkreds (m) |
| Navn | Områdets eventuelle navn |

| Jernbane.shp – jernbane | |
|--------------------------------|-------------------------------|
| <i>Tabelattributter:</i> | <i>Forklaring:</i> |
| Shape | Line (linie) |
| Type | Tomt felt, ingen definitioner |

| Vandloeb.shp – Vandløb | |
|-------------------------------|--------------------|
| <i>Tabelattributter:</i> | <i>Forklaring:</i> |
| Shape | Line (linie) |
| Typ | 16 - Vandløb |

| Transport.shp – stationer, færgehavne og lufthavne | |
|-----------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| <i>Tabelattributter:</i> | <i>Forklaring:</i> |
| Shape | Point (punkt) |
| Transpnavn | Tekstbeskrivelse |
| Type | 3 – S-tog stationer 4 – Stationer, generelt 10 – Trinbræt 15 – Lufthavne 16 – Flyvepladser 17 – Færgehavne |



| Geosted.shp – stednavne | |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Tabelattributter:</i> | <i>Forklaring:</i> |
| Shape | Point (punkt) |
| Stednavn | Stednavne |
| Type | 2 – Stednavn (henholdsvis byer) 20 – Camping 21 – Havn 22 – Lystbådehavn 23 – Golfklub 24 – Sygehus 25 – Kirker og sogn 26 – Forlystelsespark 27 – Museer og seværdighed 28 – Seværdighed 29 – Haver og park 30 – Zoologiske have 31 – Vandrerhjem 32 – Hal |

8.4 Arbejdslog

Logbog

26.03.2014

- Fremmøde
 - Alle var der - jubiii.
- Gruppen glemte at lave logbog og arbejdsblade mandag d. 24. I korte træk formåede vi at:
 - Gøre programmet færdigt og fixe bugs.
 - Gruppen dannede sit et overblik over rapportskrivningen og skrev en outline.
 - Vi brugte læringsmålene som udgangspunkt (godt tip fra Sune).
- Vi fortsatte arbejdet på rapporten
 - Læste hinandens afsnit igennem og gav nogenlunde konstruktiv feedback.
 - Vi diskuterede en hel del omkring hvad der skulle stå i hvilke afsnit.
 - Blev lidt mere fortrolige med LaTeX.
- Evaluering af produktivitet:
 - Vi var ikke særlig produktive, men fik diskuteret en masse ting.
- Evaluering af samarbejde:
 - På trods af at der et par gange i forbindelse med diskussioner blev lidt heftig stemning som følge af at gruppens deltagere var meget passionerede, gik samarbejdet fint.

19.03.2014

- Fremmøde
 - Lidt forsinket, men alle var der
- Vi arbejdede på fire forskellige ting i dag
 - Malthes arbejdede på panning. Det viste sig at være svært, og den nuværende implementation er ikke optimal.
 - Ans arbejdede på at få opdelt vinduet i to paneler, således at der nu er en label i bunden af vinduet der på sigt skal vise det nærmeste vejnavn.
 - Mark implementerede zoom vha. keyboard. Implementationen fungerer som planlagt.
 - Bjørn implementerer selection zoom. Det var svært, men implementationen fungerer som planlagt.
- Vi havde problemer med internettet. Det blev løst for Mark, Malthes og Bjørns vedkommende med ethernet kabler. Ans brugte det trådløse uden alt for store problemer.
- Vi havde igen i dag vanskeligheder med Git. Det hjalp lidt at vi endelig er begyndt at forstå det mærkelige konflikt hieroglyffer. Brugte stadig en hel del tid.
- Følgende opgaver tilbage
 - Mouse panning (delvist implementeret)
 - Show closest road (label er implementeret)

- Zoom details
- Keyboard zoom (er implementeret, men skal reagere på de rigtige taster)
- Mouse roll zoom (kan hurtigt udledes fra keyboard zoom)
- Vi har lavet en halv aftale om at mødes i morgen ~14.30 da vi ikke helt er der vi gerne ville være planmæssigt med vores implementation.
- Evaluering af produktivitet
 - Ret høj produktivitet, eftersom vi havde gang i fire forskellige ting, og lykkedes med at kombinere dem til sidst.
 - Som nævnt ovenfor sænkede forsinkelse, internet, og Git arbejdet lidt.
- Evaluering af samarbejde
 - Første dag vi allesammen skrev kode. Kæmpe fremskridt.
 - Mark og Ans undervurderer deres evne til at skride god og korrekt kode (siger Bjørn).

17.03.2014

- Fremmøde
 - Mark lidt for sent, meldte forsinkelse
 - Ans et par timer for sent, meldte forsinkelse
 - Stadig godt at der bliver meldt forsinkelser. Malthen og Bjørn startede arbejdet som planlagt
- Vi har fået refaktorert væsentlige dele af vores kode
 - Translator er skrevet om og der er blevet tilføjet hjælpefunktioner
 - Event listeners er blevet rykket ud i controller fra view
 - Fået skrevet klasse der fjerner unødig data fra datafiler
- Vi har problemer med at kortet er strukket en smule i højden lige nu
 - Vi har ikke kigget efter problemet endnu, løses næste gang
- Følgende opgaver er tilbage
 - Mouse panning
 - Mouse select zoom
 - Mouse roll zoom
 - Keyboard zoom
 - Zoom details
 - Show closest road
- Vi havde igen store vanskeligheder med Git der kostede os en time. Det er fortsat vanskeligt for os at merge branches og bare generelt få det til at gå op i en højere enhed.
 - Ville være en fordel hvis andre end Mark havde en fornemmelse af hvad Git er, sådan at vi kunne samarbejde om at løse problemerne
- Evaluering af produktivitet
 - Vi har været ret koncentrerede og der har været god produktivitet
 - Vi har taget hånd om nogle svære områder i vores kode, og refaktorert ret meget, hvilket betyder at vi ikke har fået lavet særlig meget ny funktionalitet
 - Uddelegering af opgaver er gået fint, og vi har næste hele tiden haft gang i mere end én ting ad gangen

- Evaluering af samarbejde
 - Vi havde stadigvæk lidt svært ved at forstå hvad hinanden talte om i forbindelse med Translator klassen, men det løste sig.

12.03.2014

- Ans meldte afbud eftersom han havde været involveret i et mindre trafikuheld.
- Vi startede med at skrive de ting ned vi manglede i forhold til kravspecifikationen.
- Derefter forsøgte vi at arbejde på flere ting ad gangen ud fra de nedskrevne punkter, hvilket gik ganske godt.
 - Malthé blev god til at flette konfliktende commits.
 - Vi andre blev gode til rollbacks.
- Evaluering af produktivitet:
 - Produktiviteten var meget højere i dag end ved de foregående møder fordi vi lykkedes med at uddelegere opgaver og arbejde på flere forskellige moduler på én gang. Her vil vi gerne sige tak til MVC framework.
 - Listen vi lavede i starten gav et godt overblik.
 - Vi var meget opslugte af arbejdet, da det var spændende, og var på den måde meget effektive da vi som konsekvens holdt minimale pauser.
- Evaluering af samarbejde:
 - Det gik pisse godt i dag. Ikke så meget at tilføje, når der ikke rigtig er problemer.

8.5 Arbejdsblade

| | |
|--------------------------|--|
| Opgave | |
| Hvem | |
| Hvornår | |
| Hvad skal der laves | |
| Hvordan har vi lavet det | |
| Problemer | |
| Næste gang | |

| | |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Opgave | Selection zoom |
| Hvem | Bjørn |
| Hvornår | 19.03.2014 |
| Hvad skal der laves | Selection zoom skal implementeres. Funktionaliteten fungerer således at man markerer et område ved at holde højreklik inde og flytte markøren, og når man slipper zoomes der ind på det markerede område. |
| Hvordan har vi lavet det | Brugt den MouseHandler klasse der allerede var implementeret, og brugt metoder fra Translator og klasserne Vector og Box. Implementerede først at der blev tegnet en firkant på kortet når man markede området, dernæst at der blev recentreret, og dernæst at der blev zoomet, hvilket lettete arbejdet. |
| Problemer | Nej. |
| Næste gang | |

| | |
|---------------------|-----------------------------------------------------------------------------------------------------|
| Opgave | Tilpasning af Window |
| Hvem | Ans |
| Hvornår | 19-03-2014 |
| Hvad skal der laves | Opdele Window i to paneler, så der er plads til at få fremvist "nærmste vejnavn" nederst i vinduet. |

| | |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Hvordan har vi lavet det | Opdelt Window i to paneler. Det ene panel er canvas, det andet panel er bottomPanel. Vi har brugt BorderLayout til at opsætte Window. I bottomPanel er der et label, hvor vejnavnet skal fremvises. |
| Problemer | Ikke rigtigt nogle problemer. Lidt omkring valg af layout. |
| Næste gang | Få fremvist vejnavn i label. |

| | |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Opgave | Paning |
| Hvem | Malthe |
| Hvornår | 19.03.2014 |
| Hvad skal der laves | Paning med musen skal implementeres, så der kan trækkes rundt på kortet intuitivt. |
| Hvordan har vi lavet det | <p>MouseListener Metoderne <i>mouseDragged</i> og <i>mousePressed</i> samt <i>-Released</i>, som implementeres fra henholdsvis <i>MouseListener</i> og <i>MouseMotionListener</i> interfacerne, anvendes til at observere musens opførsel.</p> <p>Når musen trækkes sendes hyppige kald til <i>mouseDragged</i> af Swing biblioteket. Disse anvendes til at udregne afstanden, som musen har bevæget sig mellem to drag-events. Denne afstand oversættes til en addering af center-koordinatet i translatoren, hvorefter kortet tegnes efter det nye center.</p> |
| Problemer | Første implementering tog udgangspunkt i et kvadratisk quadtræ men da dette ikke er tilfældet for modellen, var paningen skæv på den ene axe - kortet løb fra musen. For at udbedre dette anvendes nu musekoordinater, som er relative til modellens størrelse. Dette har dog haft den beklagelige bivirkning, at kortets bevægelse nu accelererer. |
| Næste gang | Udbedring af accelerering samt refaktoring af uoverskuelig kode. |

| | |
|--------|---------------------------------------|
| Opgave | Flyt controller-dele ud af viewpakken |
| Hvem | Malthe og Ans |

| | |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Hvornår | 17.03.2014 |
| Hvad skal der laves | View klassen Canvas indeholder listeners, som bør findes i controller-pakken. Disse skal derfor omskrives og flyttes. |
| Hvordan har vi lavet det | Klassen ResizeHandler i controller-pakken erstatter den indre klasse ResizeListener i Canvas. Denne anvender translatoren til at opdatere linjerne og derefter tegne dem. Herudover kører en timer i et halvt sekundt efter at programmet er blevet resized og aktiverer anti aliasing. |
| Problemer | Graphics objektet synes at bibeholde de gamle linjer, men dette blev løst ved at rydde objektet før der tegnes. |
| Næste gang | Flyt de resterende controller-dele ud af viewet. |

| | |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| Opgave | Translatorpusning |
| Hvem | Bjørn, Malthe, Mark |
| Hvornår | Mandag d. 17/3 - |
| Hvad skal der laves | Vi skal restrukturere koden, og forkorte unødigt lange eller unødigt komplicerede kodelinjer. Sikrer at der ikke er redundant kode |
| Hvordan har vi lavet det | Ved skrivning på 1-2 datamater. 2 par vågne øjne på hele forløbet |
| Problemer | Vores data er nu strukket ud i en høj slank form, der ikke passer til buttede Danmark |
| Næste gang | find out why |

| | |
|--------------------------|----------------------------------------------------------------------------|
| Opgave | Musen skal aktiveres |
| Hvem | Malthe |
| Hvornår | Mandag d. 17 |
| Hvad skal der laves | ikke så meget, men der er forsøgt en del |
| Hvordan har vi lavet det | Vi har indset at det nok var smartere at få lagt vores listeners i control |
| Problemer | kom ikke så meget videre med resten af opgaven |

| | |
|------------|-------------------------------------------------|
| Næste gang | Musen skal aktiveres med pan, og zoomfunktioner |
|------------|-------------------------------------------------|

| | |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Opgave | Fjern unødig data fra datafiler |
| Hvem | Malthe |
| Hvornår | 16.03.2014 |
| Hvad skal der laves | Da de originale datafiler indeholder meget unødig data, som aldrig tages i brug, skal disse omskrives. |
| Hvordan har vi lavet det | <p>DataPurger Læser en tekstfil med kommaseparerede punkter (både tal og strenge), og skriver dele af disse til en ny fil. Udvælgelsen af data afgøres med et indeksarray, hvori de ønskede kolonner optræder. Denne klasse kan både køres med argumenter (<input file> <output file> <index1,index2,index3...>) og uden argumenter. Køres klassen uden argumenter udvælges in- og outputfiler som standard datafilerne.</p> <p>EdgeData og NodeData De konstanter som indeholdte det unødige data er fjernet fra klasserne.</p> |
| Problemer | Der redigeres i det originale data. |
| Næste gang | |

| | |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Opgave | Gruppering af data |
| Hvem | Malthe, Bjørn og Mark |
| Hvornår | 12.03.2014 |
| Hvad skal der laves | Undgå at samtlige edges tegnes ved ethvert zoom-niveau. |
| Hvordan har vi lavet det | <p>Edges er nu grupperet efter deres trafikale prioritet. Disse grupper er i skrivende stund defineret som følger:</p> <ul style="list-style-type: none"> • Highways (rød) • Main roads (blå) • Paths (grøn) • Pedestrian (gul) • Naval (cyan) • Other (sort) |

| | |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Grupperingen er i datastrukturen foretaget ved at holde et quadtree array, hvis indeksering er ens med edge grupperne. Derudover tegnes forskellige edge grupper nu med forskellige farver som vist ovenfor. |
| Problemer | Vi fandt to edge-typer, som ikke er defineret i Kraks dokumentation (type 0 og type 95.) Der er dog kun 8 af disse udefinerede edges. Derudover kan det ses som et problem at grupperne skal vedligeholdes i kildekoden.. |
| Næste gang | |

| | |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Opgave | QuadTree søgning |
| Hvem | Malthe |
| Hvornår | 12.03.2014 |
| Hvad skal der laves | QuadTree klassen skal have en metode som finder den nærmeste edge til et punkt. |
| Hvordan har vi lavet det | QuadTree Metoden search(double[] point) er implementeret således, at den nærmeste edge til punktet returneres. Metoden benytter en begyndelsesrækkevidde (prædefineret som 20x20 pixels med centrum i det givne punkt), og findes edges inden for denne rækkevidde, gennemløbes disse og den tætteste returneres. Hvis ingen edges findes inden for rækkevidden fordobles denne, og der startes forfra. |
| Problemer | Da det er svært at teste QuadTree-klassen, da det kræver initialisering af Edge objekter. Derfor er det endnu usikkert om metodens køretid er acceptabel. |
| Næste gang | Implementer søgningen således at vejen nærmest markøren vises i applikationen. |

| | |
|---------------------|---------------------------------------------------------|
| Opgave | Controller |
| Hvem | Malthe, Ans og Bjørn |
| Hvornår | 10.03.2014 |
| Hvad skal der laves | Translator klasse og yderligere arbejde på view klasser |

| | |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Hvordan har vi lavet det | <p>Translator Henter edges fra model og skalerer dem til størrelsen på vinduet, og sender derefter linjer til view. Sørger også for at spejle y-koordinaterne og indstille farven på linjerne.</p> <p>View klasser En række mindre tilpasninger og tilføjelse af en metode der tegner linjerne.</p> |
| Problemer | Skagen mangler. |
| Næste gang | Bedre skalering efter bedre akser. Lige nu skaleres der kun efter x-aksen. |

| | |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Opgave | Færdiggørelse af datamodel |
| Hvem | Malthe, Ans og Bjørn |
| Hvornår | 05.03.2014 |
| Hvad skal der laves | Datamodellen skal færdiggøres, således view delen af programmet kan tegne kortet ud fra dataet. |
| Hvordan har vi lavet det | <p>Flere dele af modellens implementation var ringe eller utilstrækkelig, hvorfor flere klasser er blevet omskrevet. Derudover ønskede gruppen at bibeholde krak-kit koden i dens oprindelige form, hvilket den forhenværende implementation ikke opnåede.</p> <p>Edge og Node For at repræsentere krak dataet på en håndterbar måde, skrev gruppen en decorator klasse til EdgeData og NodeData. Disse klasser udbyggede de oprindelige klasser med funktionalitet, som var nødvendigt for at kunne indlæses i quad træet. Blandt andet har en edge i denne implementation en reference til sin egen start- og slutnode.</p> <p>Loader Færdiggjorde klassen, som før hed MappedKraKLoader, således at denne instansierer et QuadTree og populerer det med edges, som kender til deres start- og slutnode.</p> <p>QuadTree I quad træet er der ikke længere gjort brug af den indre klasse Boundary, men i stedet arrays (se Generelt.) Derudover er metoden queryRange, som returnerer samtlige edges indenfor to koordinatsæt, færdigimplementeret og testet.</p> |

| | |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Generelt I modellen er der nu gjort brug en ensartet metode at referere til koordinater på. Der anvendes et dobbeltarray, hvor [0] indeholder x1 og y2, mens [1] indeholder x2 og y2. |
| Problemer | Køretiden på instansiering af modellen er ganske ringe, men da dette kun sker ved programmets start, kan problemets omfang diskuteres. |
| Næste gang | Da datamodellen er færdig, kan arbejdet på at translater data fra modellen til viewet påbegyndes. |

| | |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Opgave | Implementation af kort-vinduet |
| Hvem | Bjørn og Mark |
| Hvornår | 03.03.2014 |
| Hvad skal der laves | Vi skal have implementeret en række klasser der tager sig af at lave et vindue og tegne kortet på det. |
| Hvordan har vi lavet det | Vi har lavet 3 klasser: Window, Canvas, Painter. Derudover har vi lavet en klasse ViewTest til at agere controller og teste koden undervejs. <ul style="list-style-type: none"> - Window er en extended JFrame der er vinduet. - Canvas er en extended JPanel der er tegnepladen som pt. fylder hele vinduet. - Painter tegner på tegnepladens Graphics object. |
| Problemer | Der er langt mere resourcekrævende at tegne stregerne med antialiasing end uden. Dette gør at vi skal tage en beslutning omkring hvorvidt dette er nødvendigt. Kortet vil blive meget mere letlæseligt, men der vil komme en del mere ventetid, hvor kortet opdaterer. |
| Næste gang | Et tekstpanel kan tilføjes, der noterer stregernes navne når musen holder over. |

| | |
|---------|------------------------------|
| Opgave | Implementation af KrakLoader |
| Hvem | Malthe |
| Hvornår | 03.03.2014 |

| | |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Hvad skal der laves | <p>Implementering af KrakLoaders to abstrakte metoder processNode og processEdge.</p> <p>Udformning af egen datastruktur til effektivt at opbevare og tilgå vejsegmenter indenfor et rektangulært koordinatsæt.</p> |
| Hvordan har vi lavet det | <p>MappedKrakLoader KrakLoader klassen extends af klassen MappedKrakLoader. Denne implementerer processNode og processEdge. Når disse metoder kaldes ved indlæsningen af kortdata, tilføjes disse til et quad tree.</p> <p>QuadTree Klasse som opretholder en quad tree datastruktur. Indeholder objekter af typen Locatable. Træets nodekapacitet er på nuværende tidspunkt fastsat til 500, men dette bør senere finjusteres.</p> <p>Locatable Interface som muliggør returnering af X- og Y-koordinater med metoderne getX() og getY().</p> |
| Problemer | <p>Vejsegmenter indeholder ikke umiddelbart et koordinatsæt for deres start- og slutsted. Dette kan dog løses ved at sammenholde et vejsegment med dets start- og slutkryds (node).</p> <p>Et vejsegment bør optræde i samtlige quads som dette beskærer. Dette er ikke let løseligt med de to koordinatsæt, som kan findes for segmentet. En mulig løsning kunne være at betragte vejsegmentets midtpunkt som dets quad-koordinat. Implementeres denne løsning må tegnefunktionaliteten omgå quadtræets data med omhu, da et segments godt kan gennemskære en specifik quad, uden at have midtpunkt deri.</p> |
| Næste gang | <p>Færdiggør quadtræet således at det indeholder samtlige vejsegmenter i krakdataet opdelt i rimelige quads.</p> <p>Implementer metoden queryRange i QuadTree klassen. Denne bør kunne finde samtlige vejsegmenter mellem to koordinatsæt.</p> <p>Skab dataklasser for vejsegmenter (edges) og vejkyds (nodes) som et let at repræsentere grafisk, efter datakilden er indlæst.</p> |

8.6 Samarbejdsaftale

Samarbejdsaftale

Ambitionsniveau

- Vi vil gerne have 7

Prioriteter

- Bjørn: Ikke mit super-primære fokus lige nu, synes de andre kurser er sværere/vigtigere.
- Mark og Ans: Vi synes det her kursus er vigtigt, og er første prioritet

Præferencer

- Mark og Ans vil gerne skrive kode, vil gerne blive bedre
- Malthe og Bjørn er okay tilfredse med deres kode niveau

Process/estimeringer

- Aftale om hvilke moduler der skal være færdige til næste uge
- Prøve at ramme en "weekly build" hvis muligt

Konflikthåndtering

- Evalueringer ved hvert møde
 - Samarbejde
 - Produktivitet

Mødetider

- Mandag: Klokken 10
- Onsdag: Klokken 13

Afbud

- Vi starter på klokkeslettet, venter ikke
- Melde afbud så tidligt som muligt
- Send en besked hvis man kommer for sent

Kommunikationsmidler

- Mark laver en mailing list
- Telefon:
 - Mark: 30648630
 - Malthe: 51883445
 - Ans: 30276607
 - Bjørn: 50872800

Versionskontrol

- Mark vil gerne stå for en github repository.
- Vi finder ud af hvordan det virker sammen

8.7 Forventningstabeller

Vector

Tabel 8.3 – Vector:Constructor

| Input data | Forventet output | Egentligt output |
|-----------------------|-----------------------|-----------------------|
| <code>vec(1,2)</code> | <code>vec(1,2)</code> | <code>vec(1,2)</code> |

Tabel 8.4 – Vector:Set

| Input data | Forventet output | Egentligt output |
|-----------------------|-----------------------|-----------------------|
| <code>vec(2,3)</code> | <code>vec(2,3)</code> | <code>vec(2,3)</code> |

Tabel 8.5 – Vector:Copy

| Input data | Forventet output | Egentligt output |
|-----------------------|-----------------------|-----------------------|
| <code>vec(1,2)</code> | <code>vec(1,2)</code> | <code>vec(1,2)</code> |

Tabel 8.6 – Vector:Add

| Input data | Forventet output | Egentligt output |
|----------------------------------|-----------------------|-----------------------|
| <code>vec(1,2)+vec(1,1)</code> | <code>vec(2,3)</code> | <code>vec(2,3)</code> |
| <code>vec(1,2)+vec(-1,-1)</code> | <code>vec(0,1)</code> | <code>vec(0,1)</code> |
| <code>vec(1,2)+vec(0,0)</code> | <code>vec(1,2)</code> | <code>vec(1,2)</code> |

Tabel 8.7 – Vector:Subtract

| Input data | Forventet output | Egentligt output |
|----------------------------------|-------------------------|-------------------------|
| <code>vec(1,2)-vec(1,1)</code> | <code>vec(0,1)</code> | <code>vec(0,1)</code> |
| <code>vec(1,2)-vec(-1,-1)</code> | <code>vec(-1,-1)</code> | <code>vec(-1,-1)</code> |
| <code>vec(1,2)-vec(0,0)</code> | <code>vec(1,2)</code> | <code>vec(1,2)</code> |

Tabel 8.8 – Vector:Multiplication

| Input data | Forventet output | Egentligt output |
|-----------------------|---------------------|---------------------|
| $\text{vec}(1,2)*2$ | $\text{vec}(2,4)$ | $\text{vec}(2,4)$ |
| $\text{vec}(1,2)*0.5$ | $\text{vec}(0.5,1)$ | $\text{vec}(0.5,1)$ |
| $\text{vec}(1,2)*1$ | $\text{vec}(1,2)$ | $\text{vec}(1,2)$ |

Tabel 8.9 – Vector:Division

| Input data | Forventet output | Egentligt output |
|-----------------------|---------------------|---------------------|
| $\text{vec}(1,2)/0.5$ | $\text{vec}(2,4)$ | $\text{vec}(2,4)$ |
| $\text{vec}(1,2)/2$ | $\text{vec}(0.5,1)$ | $\text{vec}(0.5,1)$ |
| $\text{vec}(1,2)/1$ | $\text{vec}(1,2)$ | $\text{vec}(1,2)$ |

Tabel 8.10 – Vector:Distance

| Input data | Forventet output | Egentligt output |
|----------------------------------------|------------------|------------------|
| $\text{vec}(1,2)$ to $\text{vec}(2,3)$ | $\sqrt{2}$ | $\sqrt{2}$ |
| $\text{vec}(1,2)$ to $\text{vec}(0,1)$ | $\sqrt{2}$ | $\sqrt{2}$ |
| $\text{vec}(1,2)$ to $\text{vec}(1,2)$ | 0 | 0 |

Tabel 8.11 – Vector:Translate

| Input data | Forventet output | Egentligt output |
|-----------------------------------------------------------------------------|-------------------|-------------------|
| $\text{vec}(1,2)$ from $\text{box}(0,0;2,4)$ to $\text{box}(0,0;4,8)$ | $\text{vec}(2,4)$ | $\text{vec}(2,4)$ |

Tabel 8.12 – Vector:MirrorY

| Input data | Forventet output | Egentligt output |
|-----------------------------------------------|-------------------|-------------------|
| $\text{vec}(1,2)$ in $\text{box}(0,0;1,3)$ | $\text{vec}(1,1)$ | $\text{vec}(1,1)$ |

Box

Tabel 8.13 – Vector:Equals

| Input data | Forventet output | Egentligt output |
|----------------------------|------------------|------------------|
| vec(7,21) og vec(7,21) | true | true |
| vec(7,21) og vec(21,7) | false | false |
| vec(7,21) og vec(0,-90) | false | false |

Tabel 8.14 – Box:Constructor

| Input data | Forventet output | Egentligt output |
|----------------------|------------------|------------------|
| vec(1,2) og vec(3,4) | box(1,2;3,4) | box(1,2;3,4) |

Tabel 8.15 – Box:Dimensions

| Input data | Forventet output | Egentligt output |
|--------------|------------------|------------------|
| box(1,2;3,4) | vec(2,2) | vec(2,2) |

Tabel 8.16 – Box:RelativeToAbsolute

| Input data | Forventet output | Egentligt output |
|--------------------------------|------------------|------------------|
| vec(0.5,0.5) i box(1,2;3,4) | vec(2,3) | vec(2,3) |

Tabel 8.17 – Box:Ratio

| Input data | Forventet output | Egentligt output |
|--------------|------------------|------------------|
| box(1,2;3,4) | vec(1,1) | vec(1,1) |
| box(0,0;2,1) | vec(1,0.5) | vec(1,0.5) |
| box(0,0;1,2) | vec(0.5,1) | vec(0.5,1) |

Tabel 8.18 – Box:Scale

| Input data | Forventet output | Egentligt output |
|----------------|------------------|------------------|
| box(1,2;3,4)*2 | box(0,1;4,5) | box(0,1;4,5) |

Tabel 8.19 – Box:FlipX

| Input data | Forventet output | Egentligt output |
|----------------|------------------|------------------|
| box(11,0;-3,0) | box(-3,0;11,0) | box(-3,0;11,0) |

Tabel 8.20 – Box:FlipY

| Input data | Forventet output | Egentligt output |
|----------------|------------------|------------------|
| box(0,11;0,-3) | box(0,-3;0,11) | box(0,-3;0,11) |

Tabel 8.21 – BoxProperCorners

| Input data | Forventet output | Egentligt output |
|------------------|------------------|------------------|
| box(51,-10;0,80) | box(0,-10;51,80) | box(0,-10;51,80) |