

**Universitetet i Oslo
Institutt for informatikk**

INF5261 vår 2006

Sluttrapport

Prosjekt WeGolf

Eivind Berg (eivindab (at) ifi.uio.no)

Anders Gjendem (andegje (at) ifi.uio.no)

Margrethe Store (margrsto (at) ifi.uio.no)

Terje K. L. Thoresen (tkthores (at)
ifi.uio.no)

Are Wold (arew (at) ifi.uio.no)

10. mai 2006



Innhold

1 Innledning.....	4
1.1 Bakgrunn.....	4
1.2 Problemstilling.....	4
1.3 Begrensninger.....	4
2 Golf på mobiltelefonen.....	5
2.1 Sentrale golfbegreper.....	5
2.2 WeGolf.....	6
2.3 Konkurrenter til WeGolf.....	7
2.3.1 Intelligolf.....	7
2.3.2 mScorecard.....	8
2.3.3 Mobile Golf Scorer.....	9
2.3.4 iGolfscorer.....	9
2.3.5 Wireless18.....	9
2.3.6 Oppsummering.....	10
2.4 Golf og teknologi.....	10
2.4.1 Gradvis innføring av ny teknologi - en golfers perspektiv.....	10
2.4.2 Parallell til Amish-folket.....	12
3 Hovedprosjekt: Opptegning av punktinformasjon på golfbanen.....	12
3.1 Mål.....	12
3.2 Utfordringer i forbindelse med terminalens begrensninger.....	13
3.3 Tekstlig eller grafisk presentasjon.....	14
3.4 Kartorientering.....	14
3.5 Kartskala.....	15
3.6 Relevante punkter.....	15
3.7 Opptegning.....	16
3.8 Utvelgelse av viktige data.....	17
3.9 Satellittbilder versus andre fremstillinger.....	17
3.10 Mobiltelefonen som GPS og GIS-enhet.....	18
4 Alternativt prosjekt.....	20
5 WeGolf versjon 2006: Et skritt mot utvidet virkelighet?.....	20
5.1 WeGolf som utvidet virkelighet ("augmented reality").....	21
5.2 WeGolf som hukommelsesprotese.....	21
6 Metode.....	22
6.1 Litteratursøk og bakgrunnsinformasjon.....	22
6.2 Prototyping.....	22
7 Implementasjon av hovedprosjekt.....	23
7.1 Prototypen.....	23
7.2 Programstruktur.....	23
7.2.1 Forklaring til klassediagram.....	24
7.2.2 Valg av koordinatsystem - WGS versus UTM.....	25
7.2.3 Geodetiske data – konvertering til og fra.....	27
7.2.4 Fordeling av arbeid mellom klient og tjener.....	27
7.2.5 Forklaring til skjerm bilde fra prototypen.....	28
7.2.6 Begrensninger i prototypen.....	28
7.2.6.1 Manglende oversiktighet.....	28

7.2.6.2 Begrenset funksjonalitet.....	29
7.2.6.3 Avhengighet av åpen kildekode.....	29
7.2.6.4 Ikke kompatibel med eldre mobiltelefoner.....	29
7.3 Fremtidige utvidelser.....	29
7.3.1 Dynamisk kartvisning basert på GPS-koordinater.....	29
7.3.2 Endre utsnitt og utvalg av kartdata.....	30
7.3.3 Registrering av slag og visning av tidligere runder.....	30
8 Teknologi.....	30
8.1 J2ME vs J2SE/J2EE.....	30
8.2 Virtual Machines.....	31
8.3 Konfigurasjoner.....	31
8.4 Profiler.....	32
8.5 Hierarki.....	33
8.6 WeGolf og teknologi.....	33
8.7 Utviklingsmiljø.....	34
8.8 Grafikkmuligheter.....	34
9 Samarbeid, prosess og kommunikasjon.....	35
9.1 Prosjektplanlegging.....	35
9.2 Samarbeid innad i gruppa.....	35
9.3 Kommunikasjon med WeGolf.....	36
9.4 Tidsplan.....	36
9.5 Verktøy i prosjektet.....	36
9.5.1 Integreert utviklingsmiljø - NetBeans 5.0.....	37
9.5.2 Versjonskontroll - Subversion.....	37
9.5.3 Tekstbehandler - OpenOffice.org.....	38
9.5.4 Wiki-nettsted for samarbeid - DokuWiki.....	38
9.5.5 Lynmeldinger - MSN Messenger.....	39
9.5.6 E-post.....	39
9.5.7 Mobiltelefon.....	39
9.6 Dokumenthistorie og tidligere arbeid.....	39
9.7 Hva kunne vi gjort annerledes?.....	40
10 Konklusjon og veien videre.....	40
11 Forkortelser.....	42
12 Referanser.....	42

1 Innledning

1.1 Bakgrunn

WeGolf [9] er et scorekort for golf, som du fører på mobiltelefonen. Løsningen består av to deler, WeGolf Telefon og WeGolf Web som kommuniserer med hverandre over internett. WeGolf Telefon er et program som kan kjøres på alle mobiltelefoner som har Java-støtte. Resultatet av hver spilte runde på golfbanen føres inn på mobilen, som laster de opp til WeGolf sin server. Resultatene lagres og gjøres tilgjengelig gjennom en webserver. WeGolfs brukere kan via nettleseren sin se statistikk over tidligere spilte runder.

Siste utgave av WeGolf tilbyr ekstra funksjonalitet som avstandsmåling med GPS, mulighet for å laste opp bilder tatt med mobiltelefonen og mer.

1.2 Problemstilling

Vi ønsker å generere et kart over golfbanen på brukerens mobiltelefon hvor viktige elementer er tegnet opp. Ved hjelp av informasjon om golfbanen, som fins på WeGolfs server eller som legges inn av brukeren, kan vi plote disse elementene inn på kartet. Golferen vil ha med seg en GPS-kapabel enhet, slik at mobiltelefonen til enhver tid vet akkurat hvor på kartet golferen befinner seg.

Golferen kan da ved hjelp av kartet orientere seg i forhold til hindringer, og spesielt se hva avstanden til en hindring er.

1.3 Begrensninger

WeGolf er dels en mobilapplikasjon, og mobile enheter fører meg seg spesielle utfordringer for design av brukergrensesnitt [1]. Kontaktflaten mellom bruker og system er mer begrenset, i form av blant annet liten skjermstørrelse og få, små taster. Utvikleren må også ta hensyn til at miljøet programvaren benyttes i er mindre forutsigbart enn på en stasjonær datamaskin, som er plattformen vi i størst grad har erfaring med å utvikle for. Disse begrensningene gjør at for eksempel design av

valg og navigasjonskontroller må gjøres med omhu [2].

2 Golf på mobiltelefonen

2.1 Sentrale golfbegreper

Siden det blir mye snakk om golf tenkte vi det kunne være greit å gå igjennom noen sentrale begreper.

Avstandsmerker: På ett golfhull er det som regel merker som viser avstanden igjen til greenen. Disse merkene er som regel plassert 200m, 150m, og 100m fra greenen.

Birdie: Ett slag bedre enn hullets par

Bogey: Ett slag mer enn par på ett hull.

Bunker: Sandgrop, gjerne plassert i nærheten av green.

Drive: Et slag fra utslagsstedet.

Eagle: To slag bedre enn hullets par.

Fairway: Kortklipt område mellom utslagsstedet og greenen.

Green: Området på ett hull hvor hullet er.

Handicap: System for å utligne forskjeller mellom spillere, slik at en profesjonell og en amatør kan konkurrere mot hverandre. Amatører starter med handicap 36, mens profesjonelle har som regel minus-handicap.

Hull: En golfbane består som regel av 18 hull. Hvert hull er fra 100 - 500 meter med diverse hindringer som rough, trær og bunkre.

Mottatte slag: Antall slag på en bane for en spiller basert på handicap. Korrigert for banens vanskelighetsgrad.

Par: Det antall slag man har satt opp for å nå greenen pluss to putter. For eksempel på et par 3 hull bruker man normalt ett slag inn på greenen og to putter.

Putt: Et slag som foretas på greenen.

Rough: Tykt og langt gress som er utenfor fairwayen.

Slope: Slope og rating er vanskelighetsgrad på banen som er satt av offisielt hold. Så mottatte slag blir bestemt av slope.

Stableford: Poengsystem for å bedømme en runde. To poeng får man for å spille et hull på sin par. Sin par er korrigert etter mottatte slag og handicap. Man får tre poeng for ett slag bedre enn sitt par. Har man ett mottatt slag på et par 4 hull kan man bruke fem slag for å få to poeng. Får man over 36 poeng på en runde kan man skrive seg ned i handicap.

Tee: Utslagssted.

2.2 WeGolf

WeGolf [9] er et scorekort for golf som du fører på mobiltelefonen.

Løsningen består av to deler, WeGolf Telefon og WeGolf Web som kommuniserer med hverandre over internett. WeGolf Telefon er et program som kan kjøres på alle mobiltelefoner som har Java-støtte. Etter at du har lastet ned og installert programvaren kan du føre score for opptil 4 spillere.

WeGolf regner ut mottatte slag ut fra ditt handicap, slope og rating, beregner score og stableford poeng for deg og de andre spillerne i flighten. I tillegg kan putter og/eller straffeslag på hvert hull noteres. WeGolf Web er en webside hvor du kan registrere spillere, baner og runder samt holde oversikt over rundene du har spilt.

Programmet er i kontinuerlig utvikling, noe som også er grunnen til at WeGolf ønsket kontakt med vår gruppe. I disse dager da rapporten ferdigskrives har det kommet en ny versjon som blant annet gjør det mulig å ta bilder med mobiltelefonen på golfbanen for å så publisere de på nettet med en gang. I tillegg kan man sende bilder via e-post til venner.

Regelendringene fra 1. januar 2006 gjør det lovlig å måle avstander med GPS på golfbanen. Har du en GPS i mobilen eller en GPS med Bluetooth, vil WeGolf gi deg lengden til hullet fra der du måtte befinne deg på banen. I den nye versjonen kan man nå lagre et ubegrenset antall koordinater på hvert hull, så man får vite avstanden til så mange punkter man måtte ønske fra sin posisjon på golfhullet.



*Illustrasjon 1:
Mobiltelefon med
WeGolf Telefon foran et
skjermbilde fra WeGolf
Web*

WeGolf har etter vår vurdering ikke slått helt igjennom på markedet enda. Omtrentlig antall brukere var på drøyt 1000, som ikke er all verden i denne sammenhengen, men det er voksende og de burde ha muligheter til å kapre mesteparten av det norske markedet.

WeGolf finnes i dag både i engelsk, norsk, svensk og italiensk versjon, og koster kr 395.

Screenshot of the WeGolf website showing a user profile and a detailed scorecard for a golf course. The profile includes name, address, phone, and units. The scorecard shows hole numbers, par, handicap, and scores for White, Yellow, Blue, and Red tees.

Illustrasjon 2: Skjermbilde fra WeGolf Web

2.3 Konkurrenter til WeGolf

Vi har sett nærmere på fem konkurrenter til WeGolf. Alle fem ser ut til å ha tilfredsstillende funksjonalitet for å fylle ut scorekort, beregne poeng og lignende. Forskjellene på programmene er mer hvilken ekstrafunksjonalitet de har, pris og plattform de kjører på.

2.3.1 Intelligolf

Intelligolf [10] er bestefar i familien. Det er de som har vært på markedet desidert lengst, og de har en enorm markedsandel. Werner i WeGolf nevnte noe om 90%, uten at vi vet noe annet enn at de har hundretusener av kunder i 125 land. De startet allerede i 1997 og har med det mange års forsprang på konkurrentene.

Intelligolf har fått en lang rekke med priser, blant annet "Pocket PC magazine Best Software Award 2005", så det er tydelig at de har en høy status i markedet. Programmet i seg selv ser ut til å ha mye og relevant statistikk både for PDAen og PC-applikasjonen. I databasen har de også 21.500 baner, noe som er vesentlig mer enn konkurrentene.

Screenshot of the Intelligolf mobile application showing a GPS screen for Hole 18. It displays distance from shot, stream, bunker, and oak tree locations, along with coordinates and a shot track.

Hole	1	2	3	4	5	6	7	8	9	Out	10	11	12	13	14	15	16	17	18	In	Total	
Par	4	4	4	5	3	5	3	4	4	36	3	4	4	5	4	3	5	4	4	4	36	72
Handicap	4	16	12	6	14	10	10	8	2		17	7	1	5	15	9	11	3	13			
White	269	320	342	526	190	496	150	306	337	3084	196	356	291	676	310	174	963	366	350	3208	6272	
Yellow	359	290	334	425	159	450	143	314	330	2673	186	352	303	465	302	151	963	359	361	3042	5915	
Blue	330	273	293	471	186	440	129	297	295	2029	145	352	339	430	271	140	462	334	317	2810	5499	
Red	360	260	253	445	126	407	106	249	240	2410	125	207	311	400	237	106	456	307	201	2542	4930	

Illustrasjon 3:
Intelligolf

Intelligolf er også en av de som tilbyr støtte for GPS. Den gir deg avstanden til ni punkter (200 meter, 150 meter og 100 meter fra hullet, front, bak, og midt på green, og tre valgfrie punkter) på hullet fra hvilken som helst posisjon på hullet.

Intelligolf er for øyeblikket kun kompatibel med PDA og PC (for de som ikke har en PDA, men likevel vil føre statistikk over resultater). De skal visstnok være i gang med å konvertere til mobiltelefoner, men vi har ikke sett noe til dette enda.

Det kommer i fem versjoner avhengig av hva man trenger og hvor mye man vil betale. Den billigste versjonen koster \$29.95 (har kun PDA-funksjonene), mens den dyreste koster \$49.95 og inkluderer GPS.

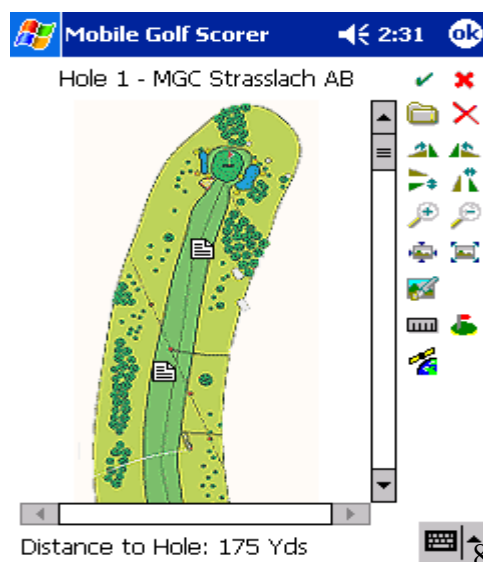
2.3.2 mScorecard

mScorecard er et relativt nytt produkt. Det ble utviklet først i 2003. Vi har inntrykk av at mScorecard [11] lider av en noe mangelfull internettside, men programmet har høstet mye ros for meget raskt og bra brukergrensesnitt, og var blant annet finalist hos Handangos beste nye applikasjon-konkurranse i 2005 (Handango [15] er verdens ledende distributør av løsninger for mobile enheter).

mScorecard har ingen støtte for GPS, og prisen er kanskje deretter. Programmet koster kun \$19.95, noe som kan sies å være billig hvis de holder det de lover på nettsidene - at gode statistikkverktøy følger med og at det er et svært raskt og lettbetjenelig program. mScorecard er også kompatibel med både mobiltelefoner, Blackberry og PDA'er.

2.3.3 Mobile Golf Scorer

Mobile Golf Scorer [12] startet utviklingen i 2003. Dette ser ut til å være blant de mest avanserte på markedet, med både integrasjon mot Google Earth, GPS og en stor statistikkmodul.



Om man har en skikkelig PDA med 640*480 oppløsning får man tilgang til en avstandsbok, som gir bilder over hull og hvor man kan definere punkter.

Mobile Golf Scorer tilbyr også GPS, som gir deg avstand fra ballen til alle koordinater du selv har lagret. I tillegg er det mulig å eksportere alle koordinatene for å gi en meget pen oversikt over runden når man kommer hjem på sin egen PC.

Programmet er kun kompatibelt for PDA og PC, og kommer i tre versjoner. Den billigste koster \$35, mens den dyreste som inkluderer GPS koster \$107.

2.3.4 iGolfscorer

iGolfScorer sier de er verdens første på GPS på mobiltelefon.

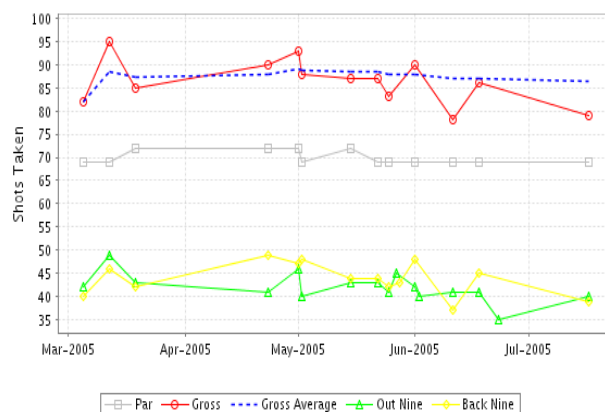
iGolfscorer [13] gir distansen til hullet fra seks punkt på hullet, altså noe færre enn de andre applikasjonene med GPS-funksjonalitet.

Det er som sagt kompatibelt med mobiltelefoner, for øyeblikket relativt nye utgaver av kun Nokia og Sony Ericsson. iGolfscorer ser ikke ut til å noe ekstern programvare, den eneste eksterne kommunikasjonen ser ut til å være å sende scorekort via e-post. Funksjonaliteten og statistikken (over maks 50 runder) som gis på mobiltelefonen ser dog ut til å være god, selv om 50 runder ikke vil holde for en golfer som virkelig vil holde oversikten.

Prisen er ikke avskrekkende. For €19.99 gir den alle funksjoner bortsett fra GPS, som du må betale €10 ekstra for.



GPS Accuracy



Illustrasjon 4: Wireless18

2.3.5 Wireless18

Wireless18 [14] har gått fra å bli en tanke hos en IT-konsulent som kjedet seg i 2003 til å bli et program med tusener av brukere i 130 land. Det har også vunnet Handango's People's Choice Award som vitner om et populært program.

Det som skiller Wireless18 fra konkurrentene er at de ser ut til å bry seg mer om samfunnet rundt spillet enn de andre aktørene. Et eksempel på dette er at de har rankinglister der alle brukerne kan

sammenligne seg selv med andre. Dette er det naturligvis frivillig å være med på. Programmet er også gratis, men alle rundene vil bli slettet fra weben etter sju dager om man ikke har betalt. Programmet er tilgjengelig på alle mobiltelefoner som støtter Java, i tillegg til PDA og Blackberry, men har ikke støtte for GPS.

2.3.6 Oppsummering

Det er nok en fordel for WeGolf at de to mest omfattende programmene ikke har støtte for mobiltelefon, og i tillegg at det ene programmet som har støtte for GPS på mobiltelefon ikke ser ut til å være spesielt imponerende (siden iGolfScorer er det eneste programmet som ikke tilbyr ekstern lagring av informasjon).

Prismessig er det stor variasjon, og WeGolf er nok blant de dyreste i klassen, men det kommer jo alltid an på hvilken funksjonalitet brukeren krever. Det er ihvertfall mange programmer som holder på å bli etablerte, og det kan være vanskelig å slå igjennom på det vi vil tro kan være et noe begrenset marked (ihvertfall i Norge).

2.4 Golf og teknologi

2.4.1 Gradvis innføring av ny teknologi - en golfers perspektiv

Golf er en veldig gammel og tradisjonsrik sport. Den har eksistert i over 500 år fra den oppstod i Skottland. Generelt har golf blitt sett på som en rikmannssport, men som nå på det nærmeste betegnes som en folkesport. Utviklingen av regelverket har gått samme vei med en liberalisering av strenge kodekser, alt fra kleskodekser til turneringsregler.

Royal and Ancient Golf Club (golfens svar på FIFA) har nå bestemt at det er tillatt å bruke hjelpemidler for å måle avstander på golfbanen også i turneringer. Det vil si at vi kanskje får se Suzann Pettersen gå rundt med en kikkert som måler avstander under turneringer snart. At GPS begynner å bli akseptert på denne måten har jo stor innvirkning på firmaer som satser på denne teknologien, som f.eks WeGolf.

Det er stadig kjente golfansikter ute og kritiserer de teknologiske fremskrittene i golfverdenen. Allerede på 70-tallet begynte man å bekymre seg, og fremskrittene til i dag har vært enorme. Det er ikke så lenge siden golf ble spilt med køller laget av tre. Så kom køllene laget av jern, og nå senest

bruker alle graffitkøller (som gjør skaftet lettere, det er bøyelig, så man får lengre baksving og høyere fart på køllehodet som igjen gir lengre slag). De siste 20 årene har også en gjennomsnittlig golfkølle blitt ca. 4cm lengre, noe som også gir lengre baksving og lengre slag. Før var golfballene laget av lær som var stappet med gåsefjær. I dag lages det golfballer som passer for alle slags spillestiler, fra baller med høy spinn til de som gir ekstra høyde eller ekstra lange slag. Det er også forskjell på baller for damer og herrer, samt at ballene har blitt stadig mer aerodynamiske.

I dag slår golfproffene mye lengre enn de gjorde før, noe som har ført til vesentlige endringer på golfbaner. En av verdens mest kjente golfbane, Augusta i Georgia USA, som holder en av fire Major-turneringer i året, har måttet gjennomgå store endringer de siste 20 årene for å holde tritt med teknologien. Hullene har blitt lengre, fairwayen blir smalere, roughen blir tjukkere og så videre.

Golfporten går gjennom en kontinuerlig teknologisk utvikling. Det blir stadig flere og bedre hjelpemidler for en jevnlig golfer, og hadde det ikke vært for en golfers stadig ønske om å ha det beste utstyret for å få de beste resultatene ville vi kanskje ikke sett samme utviklingen. Med mobiltelefon og PDA'er lager man nå programmer som gjør livet til en golfer enda lettere, spesielt med tanke på all statistikk man kan ha tilgjengelig slik at man får se hvilke steder det trengs forbedringer. Det vil alltid være noe motstand mot slikt utstyr siden mobiltelefon eller lignende ikke har noe å gjøre på en golfbane historisk sett, mens de andre teknologiske endringene vi ser er på utstyret som man bruker for å utføre sporten. Men, spørsmålet er om det fortsatt er en mobiltelefon om man bruker det til noe helt annet. Selv tror jeg at det alltid vil bli sett på som en uting. Det ideelle hadde kanskje vært om man kunne slå på telefonen og velge å bare kunne kjøre golfapplikasjonen (som man kan med MP3-spilleren på Sony Ericssons nyeste telefoner) uten å ha telefonfunksjoner. Dette kan vel riktignok bli vanskelig om det ikke er produsenten av selve telefonen som står bak applikasjonen.

Golfspillerne i dag er generelt svært negative til mobiltelefoner siden det er sett på som et forstyrrende moment. Det er et enormt irritasjonsmoment om en telefon ringer når man slår. Ernie Els, en av verdens beste spillere, foreslår at man burde bruke metalldetektorer for å luke vekk mobiltelefonen fra tilskuerne. Det vil neppe skje, men det viser at man har en fiendtlig holdning til selve objektet.

Teknologi som GPS og Wi-Fi har også fått innpass på andre områder i golfverdenen, ikke bare hos spillere. Flere klubber, spesielt i USA benytter seg nå av GPS som hjelpemiddel (GPS Industries [7] spesialiserte seg på dette). Golfklubbene har blant annet GPS på selve golfbilene (amerikanere går

jo ikke på golfbanen) for å både se avstand til hull, men også for å monitorere hastigheten i spillet. (I golf er det ofte kø for å spille, og med 10-15 minutter mellom hver gruppe som går ut på banen så trengs det bare én treg gruppe i løpet av en dag for å ødelegge hele tidstabellen). Bruksområdene er mange, for eksempel resultatvisning under turninger eller å la spillerne bruke mobilen til å få se menyer fra restauranten og bruke den til å bestille mat mens man fremdeles er på banen.

2.4.2 Parallell til Amish-folket

Howard Rheingolds artikkel om Amish-folket og dets bruk av teknologi [17] forteller om Amish-folkets skepsis til blant annet mobiltelefonen. Amish-folket er meget forsiktige med å ta i bruk ny teknologi, og de religiøse lederne har myndighet til å forby den, dersom de mener den har en negativ virkning på samfunnet. Nøkkelspørsmålet er: *"Does it bring us together, or draw us apart?"* Resultatet er en selektiv bruk av ny teknologi, hvor for eksempel telefonen er tillatt, men bare hvis den plasseres i egne telefonskur, og ikke i selve hjemmet. I hjemmet kan den forstyrre relasjonene innad i familien og trekke familemedlemmenes oppmerksomhet vekk fra hverandre. Mobiltelefonen i en golf-kontekst deler denne karakteristikken; den trekker oppmerksomheten vekk fra spillet og spillernes fellesskap.

3 Hovedprosjekt: Opptegning av punktinformasjon på golfbanen

3.1 Mål

Det er ønskelig å implementere et kart over banen hvor punkter på golfbanen som er relevant for golferen er tatt med. Dette kan for eksempel være bunkere, greenen, treklynger eller vann; objekter på banen som golferen kan orientere seg i forhold til eller bør unngå å treffe med ballen. En GPS-enhet gir data som brukes til å posisjonere golferen på kartet.

3.2 utfordringer i forbindelse med terminalens begrensninger

Det er mange problemer som kan oppstå når man ønsker å visualisere noe på en mobiltelefon. Dette inkluderer begrensninger på både hardware og utviklingsverktøy. Det finnes mye informasjon på dette feltet. Burigat og Chittaro [18], Paelke, Reimann, og Rosenbach [19], og Kray, Laasko og

Elting [20] har alle vært inne på problemer med visualisering på mobiltelefoner.

- En mobiltelefon sliter med dårlig oppløsning. De beste mobiltelefonene nå om dagen har ca. 240x320 oppløsning, mens desktopapplikasjoner har megapiksler å boltre seg i.
- En mobiltelefon har lite display.
- Telefonen kan ha et begrenset antall farger, fra få til flere tusen mot millioner av farger på vanlige PCer.
- Begrenset prosessorkraft, som ihvertfall hindrer 3D, men også animasjoner.
- Det finnes utrolig mange forskjellige konfigurasjoner på mobiltelefoner. Det er liten standardisering av brukergrensesnittet. På vanlige PCer er det både keyboard og tastatur, men for mobile enheter fins det mange unike løsninger.
- Det kan være et problem at man ikke har et fullt QWERTY-tastatur og i stedet må ta til takke med alfanumeriske tastatur.
- Det er ingen pekeenheter på mobiltelefoner, og om man har PDA må man som oftest ha egne komponenter som en penn. Berøringssensitive skjermer lider ofte av dårlig oppløsning.

Både Paelke et al. [19] og Kray et al. [20] er inne på at det slettes ikke er bare problemer med hardware som må betraktes. Applikasjoner for mobile enheter må kunne fungere i mange miljøer; også utendørs. Været kan vi ikke herske over. Kray et al. [20] skriver at sollyset var et av de største problemene for brukerne som skulle finne veien via en rutefinningsapplikasjon. Den samme artikkelen tar også opp en annen problemstilling som i høyeste grad er aktuell. De mener at brukeren bør kjenne seg igjen i sine egne omgivelser for at programmet skal være av nytte. Dette kan bli et problem for oss da vi mangler veldig mye kartdata for å få til det vi ønsker. Chittaro [22] nevner også at mobiltelefoner kan brukes i veldig mange lyssettinger, som påvirker oppfattelse av farger og grafikk.

Chittaro [22] gir en sjekklister på seks punkter som bør tenkes på før man visualiserer noe på en mobiltelefon. Vi mener tre av disse kan være viktige for oss. *Mapping*: Hvordan visualiseres sammenhengen mellom dataobjekter og visuelle objekter? Det må være en fornuftig bruk av linjer, farger, lengder, animeringer og lignende. *Selection*: På en mobiltelefon kan man ikke ha med all informasjon. Kun det viktigste bør visualiseres. *Presentation*: Er det effektiv bruk av den tilgjengelige plassen på skjermen? Selv med bra *mapping* og *selection* kan applikasjonen bli ineffektiv om ikke skjermplassen utnyttes. De andre punktene er *interactivity*, *human factors* og

evaluation.

3.3 Tekstlig eller grafisk presentasjon

Vi har fastslått at visualisering kan være vanskelig på en mobiltelefon på grunn av alle faktorer som er nevnt tidligere, så man kan spørre seg om det er hensiktsmessig i det hele tatt siden tekstbasert informasjon kan fungere like bra. Når man står på en golfbane kan man for det meste se hele hullet selv, trenger man da ytterligere visualisering på mobiltelefonen?

Masoodian og Lane [21] gjennomførte en empirisk studie av hva som var mest effektivt av grafisk og tekstlig visualisering på mobiltelefon når de skulle finne informasjon om reiseruter. Programmet ga informasjon om oppholdssted, tidssoner, datoer og klokkeslett, og forskjellen mellom den grafiske og den tekstlige versjonen var at den grafiske leverte informasjonen i grafer og diagrammer. Brukerne bestod stort sett av studenter og de ble overvåket med video for å følge med på deres handlinger da de skulle gjennomgå flere sett med spørsmål for å hente ut informasjon. Resultatet var at de som hadde den tekstlige varianten var mer effektive og fant enklere frem til informasjonen enn de som hadde den grafiske versjonen. I lys av dette kunne det kanskje være interessant å fokusere på tekstlig informasjon da dette nok er mye enklere å implementere også enn den grafiske.

Grunnen til at vi har valgt å gjøre det annerledes er også etter oppdragsgivers ønsker, samt at all informasjon han vil ha nok ikke ville gjort seg spesielt bra i tekstlig versjon.

3.4 Kartorientering

Når vi tegner opp etter GPS-koordinater vil golfbanen snu seg i forhold til himmelretningene, noe som ikke er ønskelig. Vi må altså finne en måte å snu hullet på slik at greenen alltid vil være øverst på mobilskjermen.

Kray et al. [20] var inne på at brukeren bør kunne velge å se all informasjon eller ikke. Dette er ikke et sentralt mål for dette prosjektet, men funksjonalitet for å variere hvor mye informasjon som skal vises på kartet vil være naturlig å ta med i et fullstendig program av denne typen.

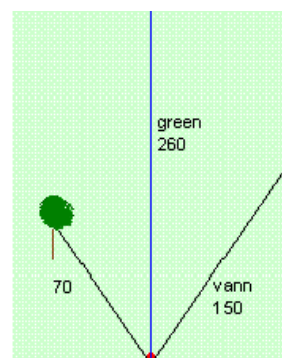
3.5 Kartskala

Utarbeiding av algoritmer for å velge ut et passende kartutsnitt å vise frem på skjermen er en sentral problemstilling. Golferen må vite om hindringene som er relevant for det neste slaget; samtidig er det ønskelig å få et overblikk over hele hullet. [3] demonstrerer en algoritme for å produsere kart velegnet for fremstilling på begrenset skjermareal. [4] viser en metode for å synliggjøre punkter som befinner seg utenfor kartutsnittet som vises. Begge artikler viser interessante metoder for å gjøre informasjon mer lettfattelig på små skjermer, og vi vil vurdere om vi kan benytte oss av en lignende metodikk.

3.6 Relevante punkter

Koordinater på golfbanen som vanligvis vil være registrert er greenen, vann, trær og bunkere. Områder, som green og vann, er oftest registrert med et frontpunkt, et midtpunkt og et bakpunkt. Vi må vurdere hva vi skal vise i slike tilfeller; det er nærliggende å tro at frontpunktet vil være mest interessant ("hvor langt unna er den vanddammen?"). Hvilke punkter som til enhver tid skal tegnes med på skjermen må vurderes nøye. Golferen må lett kunne orientere seg på hullet og forstå hvor hindringene skjermen peker ut er i forhold til ham selv. Her må vi gjøre vurderinger omkring rotasjon av kartet og hvordan vi skal posisjonere dette.

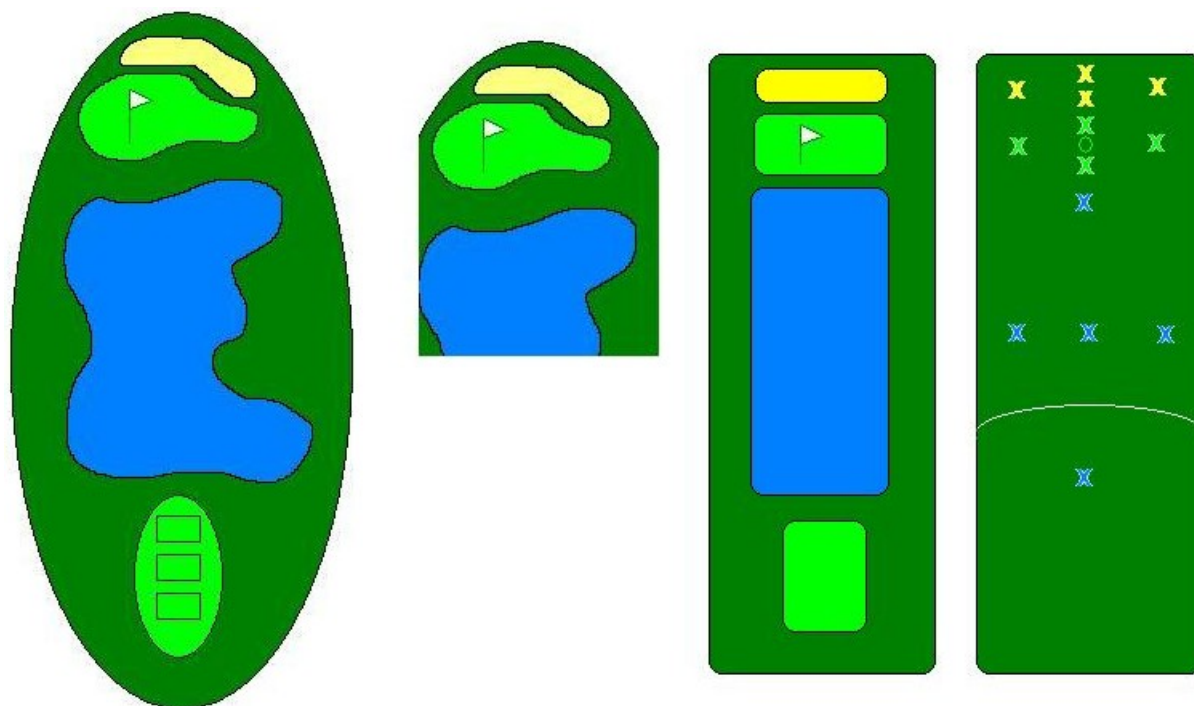
Illustrasjon 1 er et svært primitivt eksempel på hvordan det tegnede kartet kan se ut. Golferen har sin egen posisjon, retning og avstand mot greenen og en vanddam, og har en nær, synlig hindring (tree) markert for å lette orientering i forhold til de andre punktene.



Illustrasjon 5: Et eksempel på hvordan kartet kan se ut; golferens posisjon nederst

3.7 Opptegning

Vi har tenkt litt på hvordan vi skal tegne opp et golfhull på best mulig måte på en mobiltelefon. De kartdata vi har er meget begrenset, så selv om ønsket er å tegne opp mest mulig likt i forhold til realiteten blir dette umulig. Illustrasjonen under viser noen eksempler på dette. Den første tegningen viser en oversikt over hele golfhullet, som er et typisk par tre hull. Vi kunne også vurdere å tegne bare halve hullet om gangen etter hvor man er på hullet (noe som ikke er like aktuelt for denne typen hull). Dette vil begrense hvor mye som ville bli vist på displayet til enhver tid, men samtidig kunne en golfer miste noe av oversikten. Vårt tredje alternativ var en forenklet versjon av hullet som vist til høyre. Men igjen har vi faktisk ikke nok data til å lage dette, dermed blir det mest sannsynligvis versjon fire som blir det vi kan få til. Dette er som man ser en ekstremt forenklet versjon, bestående av kryss etter front, midt og bak posisjoner på hindre og green via GPS. Vi har



lagt inn flere kryss, siden vi ikke synes at front, midt og bak er nok. Legg merke til at vi ikke kan tegne opp noe før noen faktisk har lagret disse koordinatene. Et alternativ er at vi tegner opp en "dummy"-sirkel rundt dette for å få en litt bedre visualisering, noe som blir likt alternativ tre. Som nevnt tidligere strider dette mot at brukeren skal kjenne seg igjen i sine omgivelser, men det trenger ikke å være kritisk på en golfbane.

Et objekt på banen skal være registrert med minst tre koordinater, front, bak og midtpunkt. Deretter skal brukeren få vite lengden til disse punktene i forhold til hvor han står. Det kan riktignok være et problem å definere akkurat hva foran, bak og midt er for noe da dette avhenger av hvor på banen

man er. For eksempel er det mange hull som vil svinge kraftig til høyre eller venstre (såkalt "dogleg"-hull), og der vil front være forskjellig om man står på utslagsstedet eller på andreslaget. På andreslaget vil front og bak være høyre og venstre, noe som ikke er fullt så nyttig å vite. Dette er nok en grunn til at vi ønsker å registrere minst fem punkter på hvert objekt. I tillegg registreres det er punkter på banen i forhold til hvor langt man er unna flagget. Typisk er dette 200m, 150m og 100m fra hullet.

3.8 Utvelgelse av viktige data

Chittaro [22] stiller spørsmålet om hvilke data som er viktig å ta med. Han mener at å velge riktig data å vise er en av seks viktige punkter for at en skal lage en suksessful visualisering på mobiltelefon. Vi må spørre oss hvilken data om golfhullet som er viktig for brukeren å se når han står og skal slå. For eksempel er ikke avstanden til vannhinderet 50 meter til høyre noe interessant hvis man skal slå 200 meter fremover. Det er også viktig å ikke vise all informasjon på grunn av de nevnte problemene med hvor mye data som faktisk er mulig å vise på skjermen.

3.9 Satellittbilder versus andre fremstillinger

En løsning, som vi planla å jobbe med i første omgang, var om det var mulig å importere satellittbilder til telefonen, for så å kunne legge ekstra informasjon som et "overlay". Vi vurderte forskjellige løsninger for å oppnå dette:

- **Import av satellittbilder over golfbaner direkte**

Dette viste seg å ikke være realistisk, da det sannsynligvis ikke finnes noen organisasjoner som tilbyr dette gratis for kommersiell bruk. Kjøp av satellittbilder vil sannsynligvis ikke være spesielt kostnadseffektivt for WeGolf i forhold til nytteverdien.

- **Bruk av Google Earths biblioteker**

Det eksisterer et åpent språk, KML, for å kunne legge egne data som overlay i Google Earth. Det ser ikke til at det er mulig å ekstrahere selve satellittbildene til egen mobilapplikasjoner. En mulig anvendelse av KML og Google Earth er å lagre golferens slag som koordinater og så ved hjelp av KML få dette ut på Googles satellittkart. Dette er imidlertid noe som kun lar seg gjøre på web, på grunn av tilgjengeligheten av Google Earths bilder.

Det er med bakgrunn i dette at vi kan tenke oss en løsning der man tegner opp et enkelt kart over golfbanen. En mulig utvidelsesmulighet her er å sy dette sammen med fritt tilgjengelige oversiktskart fra for eksempel golfbanenes egne hjemmesider.

En annen mulig fremtidig utvidelse av hovedprosjektet kan være å hente inn en golfers tidligere runder på et hull og inkludere de på kartet over banen.

- **Bruk av tegnede kart**

En langt rimeligere løsning er å benytte tegnede kart. Enten kan vi tegne opp enkle representasjoner av golfbanene selv eller benytte oversiktsbilder av golfbanene som ofte er tilgjengelige på golfklubbenes hjemmesider. Det er denne løsningen vi ser for oss på nåværende tidspunkt.

- **Tekstlig representasjon**

En annen løsning er en mer primitiv og tekstlig fremstillingsmåte der brukeren får omtrentlig retning og avstand til forskjellige punkter på banen (bunker, freeway, rough og så videre...). Det vil være hensiktsmessig å starte med en slik representasjon, da vi på denne måten får kontrollert at klassifiseringen av de ulike områdene fungerer. Vi betrakter dette som en nødløsning.

3.10 Mobiltelefonen som GPS og GIS-enhet

Etterhvert som mobiltelefonene har blitt kraftigere og fått større skjermer, har et annet applikasjonsområde som tidligere var forbeholdt applikasjons-spesifikke enheter blitt tilgjengelig for mobiltelefonen, nærmere bestemt bruk av mobile terminaler som GPS. Ifølge artikkelen "From GIS to LBS"[8], har såkalte GIS-systemer (Geographical information systems) stort sett vært relevant for PC. Man kan tenke seg mange forskjellige bruksområder for en "mobil GIS-utgave", ikke minst WeGolf der et *utopia* vil være et system som på mange måter er like funksjonelle som et vanlig GISsystem. Blant annet kan man tenke seg en rik applikasjon der man har tilgang til ikke bare oversikt over golfbanene men også flere "layers" som inneholder informasjon om elevasjon, bakkeforhold og annen informasjon som er relevant for en golfspiller. På den annen side er det trolig bare proffe golfspillere som virkelig ville hatt nytte av slike data, og i



tillegg ville dette kreve dyre og multispektrale satellittbilder. Vi betrakter derfor slik funksjonalitet som et endelig utopia. For vår del er imidlertid enkel GIS-funksjonalitet som klassifisering av punkter eller områder relevant for oss, for å kunne gruppere de forskjellige elementene man finner på en golfbane, og vil derfor konsentrere oss om dette. Mobiltelefonene har enten innebygd GPS-enhet, eller man kommuniserer via Bluetooth med ekstern GPS-enhet, som vist på bildet.

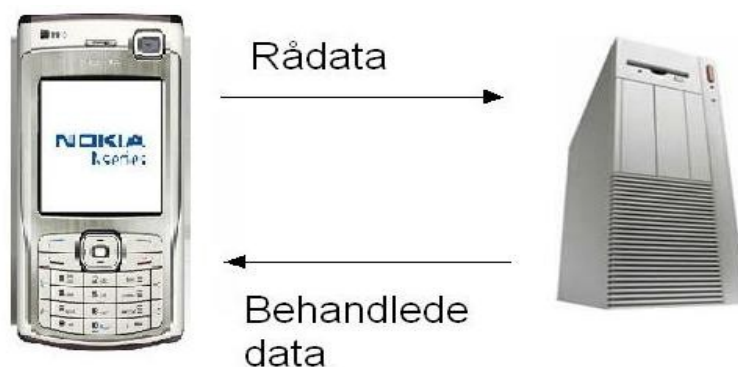
Videre må vi ta hensyn til de forskjeller mellom "vanlige" GIS systemer og mobile GIS-systemer, relatert til *Distribuerbarhet, perspektiv og mobilitet*. Disse forskjellene skiller det som omtales som vanlige GIS-systemer fra såkalte LBS (Location-Based Services).

- Mobilitet

Der vanlige GIS-systemer er basert på et overordnet bilde av ofte store landflater er LBS systemer basert på brukerens perspektiv i forhold til de *nærmeste* omgivelsene. Det er for oss uinteressant å ta med elementer utenfor golfbanen. I tillegg må representasjonene av golfbanene tilpasses de vesentlige mindre mobilskjermene.

- Distribuerbarhet

Mobiltelefoner har vesentlig mindre kapasitet enn en vanlig PC med tanke på lagringsplass, minne og prosessorhastighet. For å kunne tilby selv grunnleggende GIS-funksjonalitet kan det fort bli nødvendig å ty til en distribuert arkitektur der mobiltelefonen sender data til en server over mobilnett for å utføre mer krevende beregninger. Dette bør ikke bli et problem for WeGolf, da noe av eksisterende funksjonalitet er basert på en slik arkitektur.



- Perspektiv

Henger sammen med mobilitet. Siden golfbaner er relativt store, bør man trekke mobilitetsprinsippet enda lengre ved å tilby visning av den delen av golfbanen brukeren befinner seg i. Fra brukerens perspektiv vil dette oppleves som å "zoome" inn på den relevante delen av banen.

4 Alternativt prosjekt

Tidlig i prosjektarbeidet vurderte vi en annen problemstilling innen utviklingen av WeGolf: Å designe en IM-klient for mobiltelefonen som er i stand til å kommunisere med et tenkt diskusjonsforum under en golfers side på WeGolf sin server. Da kan man se for seg at venner og bekjente følger runden på nett og sender beskjeder til golferen gjennom WeGolf-applikasjonen. Golferen kan svare tilbake gjennom applikasjonen ved anledning.

Da dette prosjektet ville fått et større fokus på serversiden, vurderte vi det som mer interessant å prioritere arbeid med opptegning av kart på mobilterminalen. Dette har ført til at vi har gjort en del praktisk programmering i J2ME, og vi har stiftet bekjentskap med problemstillinger vi hadde møtt på i mindre grad dersom vi hadde satset på det alternative prosjektet. Eksempler på dette er konstruksjon av gode grafiske representasjoner av geografiske data på en liten skjerm og øvelse i J2ME-APIet for å tegne 2D-grafikk.

5 WeGolf versjon 2006: Et skritt mot utvidet virkelighet?

Artikkelen "Just say Nokia" [25] tar for seg finske forskeres syn på hvordan mobiltelefonen kan utvikle seg til å bli en enhet som understøtter utvidet virkelighet, eller "augmented reality":

"On a train-station platform, a line of text visible in your data glasses informs you the train is 15 minutes late; more text captions flash over the heads of others on the platform, telling you who they are or who's just around the corner."

En kombinasjon av sensorer og visningsenheter man kan ha med seg - og ha på seg - muliggjør en sammensmelting av den virkelige verden med data som kommer fra mobiltelefonen.

Videre ser forskeren for seg at mobiltelefonen på sikt kan bli enheten som tar opp alt du ser og hører - en slags hukommelsesprotese.

5.1 WeGolf som utvidet virkelighet ("augmented reality")

Med fungerende opptegning av kart på mobiltelefonen har vi tatt et skritt mot å bygge en utvidet virkelighet for golfspilleren, som da kan se virkeligheten rundt seg gjenspeilet på skjermen på mobiltelefonen. Takket være bruken av GPS-mottaker vil de to "datakildene" (virkeligheten og mobiltelefonens skjerm) være synkronisert.

På sikt kan vi se for oss at skjermen på mobiltelefonen, som allerede i dag fungerer som en søker når mobiltelefonen brukes som kamera, kan benyttes til å få informasjon om miljøet rundt spilleren. Golferen ser virkeligheten rundt seg gjennom kameratelefonens skjerm, og informasjon om distanser, tidligere slag og så videre legges over bildet som kameraet på mobiltelefonen fanger inn.

5.2 WeGolf som hukommelsesprotese

Dagens versjon av WeGolf fungerer allerede som en begrenset hukommelsesprotese. Ved at golferen fører resultatene inn i programmet, lastes informasjonen opp til serveren, som gjør informasjonen tilgjengelig i form av grundig statistikk om spillet. Denne kan hentes frem ved senere anledninger - både for å se hvordan akkurat *den* runden gikk, og for å vurdere sitt eget spill over tid.

I den aller siste versjonen av WeGolf er det i tillegg til resultatinformasjon mulig å laste opp bilder som tas med kameraet på mobiltelefonen, slik at man kan lage en blogg med bilder, hvor man skriver inn teksten i etterkant gjennom web-grensesnittet. Bildene knyttes også til scorekortet fra den aktuelle runden. Dermed har man muligheten til å se akkurat hvordan man var kledd, hvordan været var og så videre - i tillegg til eksakt hvilket resultat man oppnådde. Programvaren sørger for å ta vare på stadig mer informasjon om golferens opplevelse. Vi ser for oss at dette vil bli mer og mer omfattende i fremtiden; i første omgang vil videoklipp være en naturlig utvidelse (kompisen kan filme teknikken på slaget, som blir lagt opp og koblet med statistikken). I et lengre perspektiv - når "video glasses" ("videobriller") eller en tilsvarende opptaksenhet har blitt en realitet - vil man kanskje filme hele golfrunden fra førsteperspektiv, og "hukommelsesprotesen" som Nokia-forsker Nieminen diskuterer [25] er mer eller mindre blitt virkelighet.

Problemstillingen er: Hvilken verdi gir dokumentasjonen som produseres? Hva er den rette balansen mellom å ta vare på de øyeblikkene som man virkelig vil glede seg over i ettertid, og å produsere timesvis med uinteressante opptak? Statistikken som WeGolf samler i dag er tall, som lett lar seg aggregere og søke i. Tilsvarende teknologi for lyd og bilder er på et svært tidlig stadium.

6 Metode

Vi har ikke benyttet et spesielt metode-rammeverk i dette prosjektet. Gruppen har hatt flere møter med oppdragsgiver, hvor vi har diskutert frem en problemstilling og grov arbeidsplan. Deretter har vi forsøkt å fordele arbeid innad på gruppen. I forhold til koden vi har ønsket

- Kartlegging av krav
- Modellering
- Implementering og testing

I praksis har kodingen skjedd i forhold til målet i hovedprosjektet, og med liten grad av planlegging og modellering på forhånd.

6.1 Litteratursøk og bakgrunnsinformasjon

For å få litt bakgrunnsinformasjon og forståelse av problemer med tegning av kart på mobiltelefonen foretok vi litteratursøk i blant annet Google Scholar [30] og Informatikkbiblioteket[29].

Vi ble gjort oppmerksom på at det finnes konkurrerende programvare; undersøkelsen av de største aktørene i dette markedet har gitt oss nyttige innspill til vårt eget prosjekt.

6.2 Prototyping

Grunnet den relativt korte tiden prosjektet går over bestemte vi oss for å konsentrere oss om å lage en prototype og ikke implementere en ferdig versjon. Prototyping tillater oss å eksperimentere med forskjellige tilnærminger til problemet og lære mest mulig av prosjektarbeidet. I en komplett implementasjon vil mye tid gå med til programkoden og planleggingen som trengs for å lage en

fullstendig løsning. En prototype kan eventuelt være et grunnlag for videre arbeid med problemstillingen.

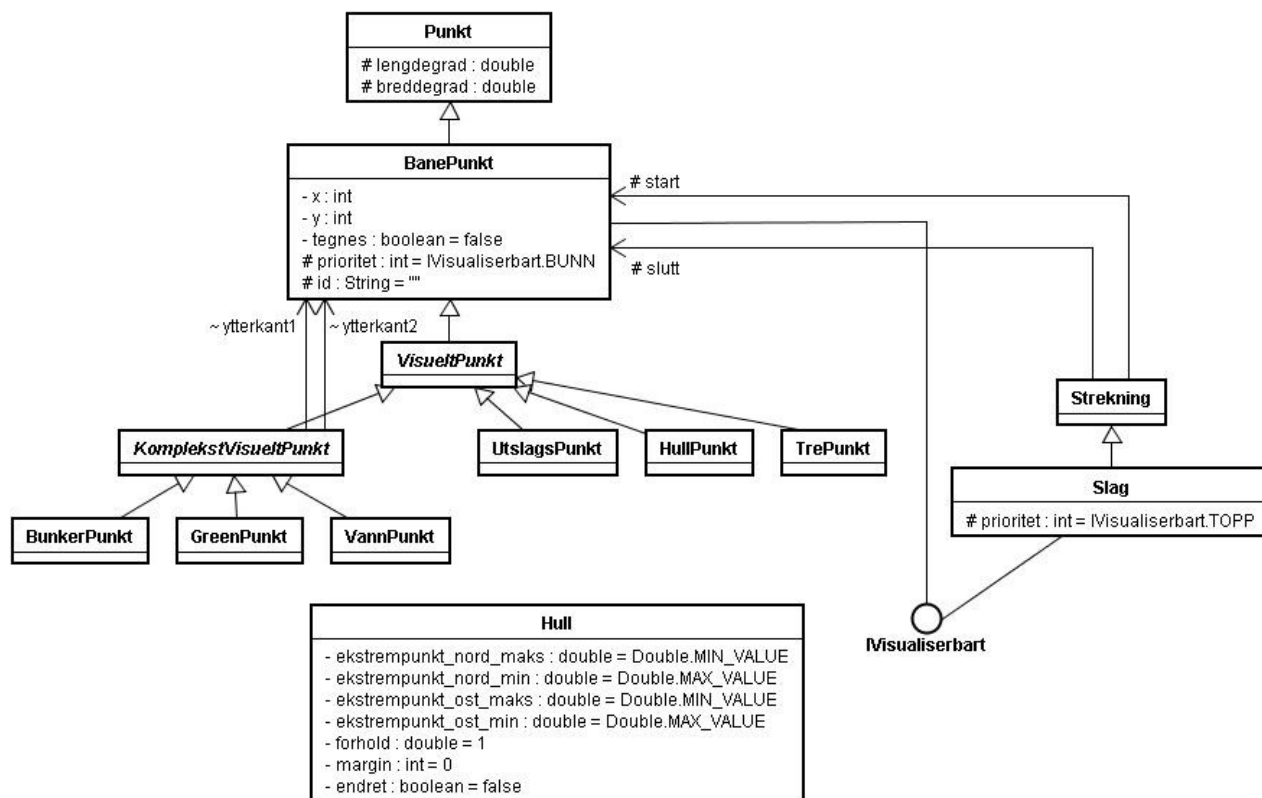
7 Implementasjon av hovedprosjekt

7.1 Prototypen

Vi har implementert en enkel prototype som er i stand til å tegne opp et bilde av golfbanen. Programmet er per i dag ikke knyttet opp til GPS-signalet og kan dermed kun tegne opp et statisk situasjonsbilde gitt en samling koordinater.

Forbindelse til serveren (hvor kartdataene vil hentes fra) mangler, da dette ikke er nødvendig for å lage en implementasjon av karttegningsdelen av programmet. For WeGolf er det også en vurdering i forhold til å gi oss tilgang til deres kildekode, noe som de på denne måten fullstendig unngår.

7.2 Programstruktur



Illustrasjon 6: Klassediagram i UML for prototypen

7.2.1 Forklaring til klassediagram

Klassen som implementerer brukergrensesnittet (og som ville tatt seg av kommunikasjon med server om det var implementert) i programmet er **GolfGUI**, og er ikke tatt med i UML-diagrammet.

Klassen **Hull** representerer ett hull på en golfbane, og instansieres av **GolfGUI**. **GolfGUI** oppretter alle objektene hullet skal fylles med (klassene **BunkerPunkt**, **GreenPunkt**, **VannPunkt**, **UtslagsPunkt**, **HullPunkt**, **TrePunkt** og **Slag**).

Områder (**BunkerPunkt**, **GreenPunkt** og **VannPunkt**) arver en klasse kalt **KomplekstVisueltPunkt**. Forskjellen på denne og et **VisueltPunkt** er at et **KomplekstVisueltPunkt** er et område angitt av ett **VisueltPunkt** og to **BanePunkt**, som typisk angir midtpunkt, forkant og bakkant på et område.

UtslagsPunkt, **HullPunkt** og **TrePunkt** angir enkeltpunkter på hullet (**HullPunkt** er selve koppen; denne flyttes vanligvis mellom runder, og dette koordinatet vil dermed normalt ikke være i bruk).

Disse tre klassene i hver gruppe er svært like; bare deres metode for å tegne seg opp er forskjellig.

Punkt inneholder de reelle koordinatene i verden for punktet. **BanePunkt** har i tillegg int x,y som angir hvor på punktet tegnes på en skjerm. *Prioritet* angir hvorvidt punktet skal tegnes foran eller bak andre punkter. Tekststrengen *id* brukes til debugging.

Slag representeres av to **BanePunkt**.

7.2.2 Valg av koordinatsystem - WGS versus UTM

Når man bruker mobiltelefonen som GPS-enhet, kommer man ikke utenom noe kunnskaper om koordinatsystemer og hvilke konsekvenser som følger bruk av forskjellige koordinatsystemer.

Innen navigasjon eksisterer begrepet "datum", som er definert som "en referanse hvorfra målinger av jordoverflaten har blitt foretatt". Innen kartlære er dette et punkt på jordoverflaten som er referanse for andre oppmålinger, slik at man kan lage en "modell" av jorda. Det finnes mange datum, for flere forskjellige bruksformål. En rekke "lokale" datum eksisterer, og disse er primært ment for å dekke en mindre del av jordoverflaten. For å dekke større deler av overflaten, bruker man "geodetiske" datum som for eksempel WGS84 (World Geodetic System), som brukes som det primære koordinatsystemet i GPS, dette fordi avstander mellom posisjoner vil være relativt nøyaktig i dette datumet.

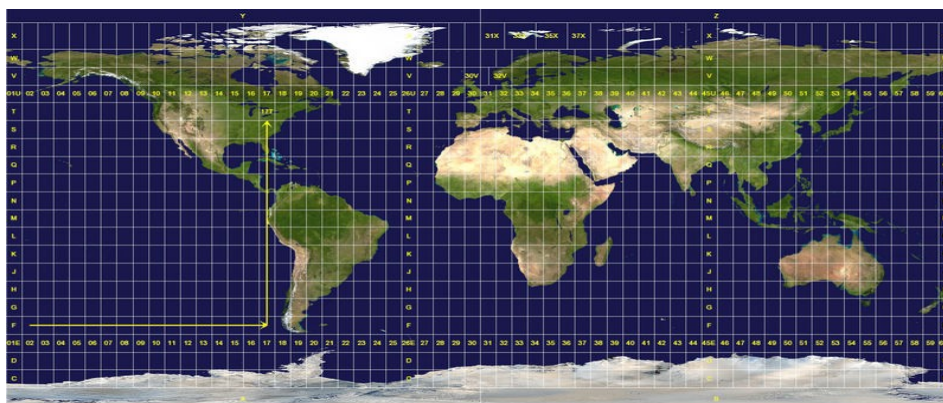
(GPS-enheter understøtter for øvrig ofte andre datum i tillegg til WGS84, men sistnevnte er mest utbredt.)

Fordi WGS84 brukes som standard GPS, vil systemet selvsagt være et naturlig valg i den posisjonsbaserte delen av WeGolf. Imidlertid blir opptegning av elementer på skjermen noe vanskeligere, fordi man må ta hensyn til økende konvergeringsgrad av lengde -og breddegradsceller jo lenger nord man kommer ved å bruke matematiske formler. For eksempel vil slike "celler" være mer rektangulære ved ekvator, men i svært liten grad nær nordpolen. Slike matematiske formler er vanligvis tilgjengelige for Java på desktopsystemer (J2SE), men man finner ofte ikke igjen sentrale funksjoner som `aCos()` i Java for små enheter (J2ME).

Med andre ord er det relativt enkelt å beregne *avstander* i WGS, ved hjelp av trigonometriske formler, mens *retninger* og *elementer* man vil tegne opp på skjermen er mer komplisert grunnet innsnevringen av koordinatcellene.

For å enklere kunne forholde oss til plotting av punkter på (den rektangulære) mobilskjermen, har en endring av prototypen bestått i å bruke UTM (Universal Transverse Mercator) systemet istedet.

Fordi UTM fra et praktisk synspunkt i det store og hele består av et rektangulært rutenett, er det relativt lett å beregne både avstander og retninger, ved hjelp av for eksempel Pythagoras. Grunnen til at UTM muliggjør rektangulære beregninger istedet for å måtte bruke beregninger som tar hensyn til kurver, er blant annet at UTM ikke består av en enkelt *projeksjon* som dekker jordoverflaten, men er satt sammen av mindre projeksjoner som utgjør et rutenett, som vist nedenfor. En projeksjon er et perspektiv hvorfra man kartlegger en del av (eller hele) jordoverflaten.



UTM-soner

Ulempen er at GPS bruker WGS84, og vi vil da måtte konvertere vårt sett av koordinater til UTM-koordinater på en eller annen måte, og en kan gjøre dette på hovedsakelig to måter: Enten måle opp referansepunkter manuelt med GPS-enhet som støtter UTM, eller konvertere til UTM i programvare. Sistnevnte alternativ er det mulig å implementere selv, men god innsikt i navigasjon er nødvendig. Alternativt kan tilgjengelige Java-biblioteker benyttes. Vi har funnet noen slike, men problemet her har enten vært at disse ikke er tilgjengelige under lisenser som muliggjør bruk i lukket, kommersiell programvare eller at de bruker standardbiblioteker som kun finnes i J2SE. Disse kan med andre ord ikke benyttes på en J2ME-enhet.

"Konverteringskalkulatorer" (som er tilgjengelig på nettet) kan brukes til konvertering fra WGS84 til UTM, men dette vil medføre mye arbeid ved store mengder koordinater. Konverteringen vil måtte gjøres for hver gang brukeren legger til en posisjon på golfbanen (for eksempel et utslagspunkt), men dette bør ikke medføre regneproblemer på nyere mobiltelefoner, selv med et signifikant sett med punkter som skal konverteres.

7.2.3 Geodetiske data – konvertering til og fra

Som nevnt tidligere finnes det noen applikasjoner/biblioteker man kan integrere i Java-applikasjoner, for vår del er det mest interessant med konverteringsmuligheter mellom WGS84 og UTM.

Etter hva vi har erfart er mange av disse ikke gratis for kommersielle formål, og funksjoner tilgjengelige i biblioteket kaller ofte matematiske J2SE funksjoner for å foreta sine beregninger. Selv om man da kunne implementert bibliotekene på en enhet med J2ME, så ville ikke funksjonskallene i biblioteket fungere, da J2ME kun har et subsett av de matematiske funksjonene i forhold til J2SE. Man må i så fall implementere matematikkfunksjonene selv.

Man finner imidlertid noen biblioteker/applikasjoner under GPL, men mange av disse er kun implementert for bruk med J2SE, og kvaliteten og dokumentasjonsgraden vil nok være varierende da disse ofte er implementert av privatpersoner.

Videre eksisterer det mye Javakode i form av hele applikasjoner. Vil man integrere funksjoner fra disse applikasjonene (dvs, man benytter kun metoder, ikke applikasjonen i sin helhet) vil det være en forutsetning at de er distribuert under GPL slik at man har tilgang til dokumentasjon og kildekode. Imidlertid tillater ikke GPL at man publiserer GPL-kode som en del av en applikasjon som ikke publiseres under GPL; da må man finne alternativer som er publisert under Lesser GPL, for eksempel, som tillater slik bruk.

Et eksempel på Java-applikasjon fra privatperson kan sees på <http://www.i3s.unice.fr/~johan/gps/> [34]. Applikasjonen er gitt ut av en privatperson under GPL, og kildekode som støtter konvertering fra WGS84 til UTM er tilgjengelig. Dokumentasjon til koden er tilsynelatende ikke lett tilgjengelig, derfor er dette biblioteket foreløpig kun interessant som en mulig fremtidig måte å foreta konvertering på.

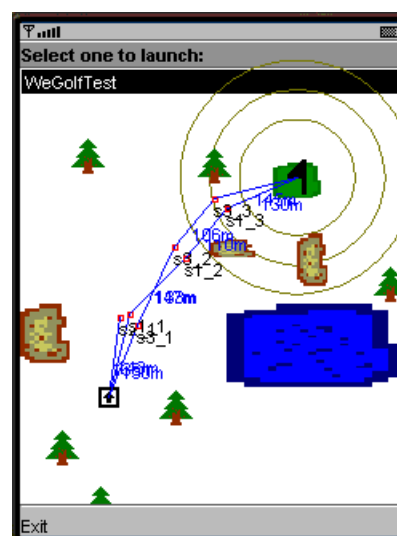
7.2.4 Fordeling av arbeid mellom klient og tjener

Prototypen gjør alt arbeidet med prosessering av kartinformasjon på klientsiden (selv hvis vi hadde implementert forbindelsen med serveren). Dette sparer datatrafikk, slik at man slipper å vente på at serveren skal gjøre utregninger og unngår kostnadene datatrafikk medfører. Imidlertid medfører dette forholdsvis store krav til maskinvaren på klientsiden, som må støtte CLDC 1.1 og være rask nok til å besørge en god brukeropplevelse.

Alternativet til å bygge kartet på klienten er å la serveren gjøre jobben med å regne om høyde / bredde-koordinater på elementer på golfbanen til en enkel datastruktur, ta imot lokasjonsinformasjonen produsert av brukerens GPS-enhet, og sende en optimalisert utgave av informasjonen som trengs for å tegne kartet til klienten. Klienten vil i et slikt scenario bare ha enkel kode for å tegne opp kart samt kommunisere med serveren.

7.2.5 Forklaring til skjermbilde fra prototypen

Skjermbildet viser et kart over et tenkt golfhull. GPS-posisjonene er ekte, da vi tok en tur med en GPS og lagret en del koordinater. Så i praksis vil et golfhull være en del mindre enn dette da det nå er ca. 700 meter langt. Alle hindringer er tatt med og ingenting er zoomet, derfor er det en del plass på skjermen som for øyeblikket er bortkastet.



Illustrasjon 7: Prototypen kjøres på en emulator via NetBeans

Spillerens posisjon er den svarte firkanten med pila. De blå strekene markerer ulike slag på dens vei til greenen. Markert på skjermen er ulike hindringer som bunkere og trær. Sirklene er standard avstandsmåling som finnes på de fleste golfbanene. Her har vi markert hvor det er 100m, 150m, og 200m igjen til hullet.

For øyeblikket vises det i standard himmelretninger, som vil si at nord alltid vil være øverst på skjermen. I utgangspunktet ville det mest logiske være at greenen alltid var øverst, men dette er ikke gjort foreløpig.

7.2.6 Begrensninger i prototypen

Her vil vi redegjøre for nevneverdige begrensninger i prototypen.

7.2.6.1 Manglende oversiktighet

Dagens prototype blir veldig rotete når man skal vise mange slag. I og med at en spiller kan ha svært mange tidligere slag på det samme hullet vil man ha problemer med en overfylt skjerm. Her har Chittaro [22] som nevnt tidligere et viktig poeng at man må velge ut den viktige dataen. Historisk sett er det nok ikke viktig å vise alle de tidligere slagene man har på et hull, men kanskje heller for de ti siste rundene så man får se en tendens.

Alle slag er for øyeblikket markert med en meterverdi i tekst, noe som bør fjernes ved flere slag.

7.2.6.2 Begrenset funksjonalitet

Siden prototypen kun produserer et statisk bilde og mangler støtte for interaksjon med brukeren eller GPS-enheten, får vi ikke testet den i praktisk bruk. Vi kan imidlertid gjøre vurderinger av hvor lesbart vårt design er på en mobil terminal, og sammenligne våre konkrete erfaringer med andre som har jobbet med kartdata på små enheter.

7.2.6.3 Avhengighet av åpen kildekode

Kode for utregning av avstander mellom punkter er forholdsvis kompleks å skrive, og prototypen benyttet i en periode åpen kildekode. Denne kildekoden er publisert under en lisens som ikke uten videre tillater at den benyttes i en kommersiell applikasjon. Denne koden må med andre ord lisensieres eller erstattes med annen kode før den kan benyttes i WeGolf. Dette var en av grunnene til at vi gikk fra å benytte WGS84-formatet til UTM.

7.2.6.4 Ikke kompatibel med eldre mobiltelefoner

Prototypen krever at den mobile enheten støtter CLDC 1.1 på grunn av utregninger med desimaltall. Dette betyr at en betydelig andel av mobilterminalene på markedet ikke er i stand til å kjøre prototypen. Hvorvidt dette er et problem spørs på om disse eldre mobiltelefonene utgjør en målgruppe for applikasjonen. Dagens WeGolf-applikasjon krever kun CLDC 1.0, men en rekke forbedringer man finner på nyere telefoner, som bedret skjermopløsning, fargeskjerm og mulighet for trådløs kommunikasjon med GPS-mottakere, gjør at fremtidens WeGolf-applikasjon sannsynligvis uansett vil bli skrevet for telefoner hvor CLDC 2.0 er inkludert.

7.3 Fremtidige utvidelser

7.3.1 Dynamisk kartvisning basert på GPS-koordinater

Det naturlige neste steget er å utvide programmet til å ta imot koordinater fra GPS-senderen og justere fremvisningen av kartet i henhold til disse, slik at vi får et dynamisk bilde av golfbanen som endrer seg ettersom golferen beveger seg rundt på banen.

7.3.2 Endre utsnitt og utvalg av kartdata

På kart med høyt detaljenivå vil det være en fordel å kunne zoome inn på kartet, for å gjøre bildet mer oversiktlig. Det er også vanskelig å forutsi hva slags kartutsnitt den enkelte bruker vil foretrekke; dermed er det fordelaktig å la brukeren selv kontrollere dette.

Tilsvarende vil det være en fordel å la brukeren ha innflytelse over hvilke koordinater som tas med på kartet, slik at golferen selv kan styre hvor mye data som vises på kartet.

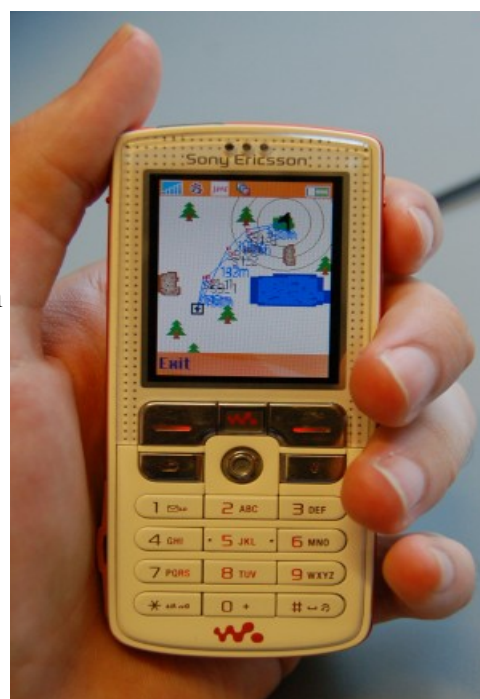
7.3.3 Registrering av slag og visning av tidligere runder

Ettersom runden gjennomføres kan golferen registrere at et slag ble gjort. Databasen vil dermed etterhvert ha en serie koordinater som angir når et slag ble gjort og hvilken posisjon det ble slått fra. Denne serien vil vise hvordan golferen beveget seg over banen.

En slik koordinatserie kan vises på skjermen, i tillegg til de andre kartdataene, slik at golferen kan se hvordan han tidligere har spilt på denne banen og ta dette med i beregningen når han gjennomfører runden.

For å lage denne funksjonaliteten må vi implementere kode for å lagre og hente frem slike koordinater på serversiden og implementere kommunikasjon med serveren på klienten.

Prototypen gjør nå litt av denne jobben; den tegner opp en serie slag, men visningen av disse kan ikke slås av, og koordinatene er dummyverdier.



Illustrasjon 8: Prototypen kjøres på en Sony Ericsson W800i

8 Teknologi

8.1 J2ME vs J2SE/J2EE

Sun har valgt å lage forskjellige utgaver av Java tilpasset forskjellige typer omgivelser. På mobiler

er det vanlig å bruke J2ME – Java 2 Micro Edition, en utgave spesielt tilpasset enheter med begrensede ressurser. I praksis betyr dette for utvikleren at man har et relativt begrenset API å forholde seg til for å skrive koden, og at mange av de klassene og metodene de fleste Javautviklere er kjent med fra den vanligste versjonen, J2SE (Java 2 Standard Edition) rett og slett ikke eksisterer. Kombinasjonen av begrensede ressurser og manglende APIer gjør utvikling for enheter som bruker J2ME til en utfordring i mange situasjoner. Et eksempel på dette er at flyttall mangler på mange enheter, og man kan risikere å måtte lage metoder for å konvertere heltall til flyttall for visning på skjerm selv ved hjelp av tekstmanipulasjon.

8.2 Virtual Machines

Enheter som bruker J2ME har på grunn av sine begrensede ressurser også en annen Virtual Machine enn J2SE/J2EE (Disse bruker normalt HotSpot VM (Virtual Machine) i dag – "fullversjonen"), mens J2ME bruker KVM – Kilobyte VM. En del "mellomstore" enheter bruker en mellomting – nemlig CVM – Compact VM, eksempel på dette er Set-Top-bokser og enkelte PDA'er. Foreløpig har vi altså 3 forskjellige versjoner. Problemet Sun da stod ovenfor var at de ikke ville begrense J2ME helt ned til laveste felles nevner. Løsningen ble å innføre forskjellige konfigurasjoner og profiler under J2ME igjen.

8.3 Konfigurasjoner

En konfigurasjon definerer hvilke muligheter som skal være tilgjengelig. En konfigurasjon brukes på en hel serie av produkter med tilsvarende funksjonalitet og behov. Konfigurasjonen definerer for eksempel minimum maskinwarespesifikasjon, som minne, skjerm og så videre, samt hvilke bibliotek som skal være med.

CDC: Connected Device Configuration

- Set-top-bokser og PDA'er, bruker CVM.

CLDC: Connected Limited Device Configuration

- Stort sett mobiler og noen PDA'er. Denne er også kjent som "You-wanna-run-Java-on-what?!"

For å gjøre det enda litt mer detaljert er disse igjen oppdelt i versjoner. CLDC finnes for eksempel i

utgave 1.0 og 1.1 – sistnevnte konfigurasjon støtter for eksempel flyttall som nevnt over. CLDC 1.1 er vanlig på de fleste nyere mobiler i dag, men det er svært mange ute i markedet fremdeles som ikke støtter mer enn versjon 1.0.

8.4 Profiler

En profil er igjen en slags utvidbar underkategori av en konfigurasjon. CLDC sier ingenting om GUI, det gjør profilen i vårt tilfelle; MIDP – Micro Information Devices Profile. Profilen er det laget som legger til kategori-spesifikke komponenter på toppen av en konfigurasjon; med andre ord er MIDP mer direkte rettet mot mulighetene mobiltelefoner har. MIDP definerer igjen en minste felles nevner for hva disse enhetene skal tilby av hardware og kommunikasjonsmuligheter samt minimumsspesifikasjoner for disse. (Tastatur, nettverk, minne (Temporært minne, ikke-volatilt for f.eks. konfigurasjon av programvaren/settings og for selve programvaren som kjøres), skjerm)

I tillegg er det MID-profilen som faktisk tilbyr bibliotekene man programmerer mot.

MIDP-program har begrenset tilgang til ressursene på enheten. Alle ressurser som skal brukes må være tilgjengelig via CLDC og/eller MIDP-bibliotekene, i JAR-filen som programmet selv kom i, eller via programmets beskrivelsesfil – JAD-filen. JAR-filen er en utvidet ZIP-fil på samme måte som i vanlig J2SE, den inneholder de kompilerte klassene, og eventuelle andre ressurser programmet trenger som bilder og tekst-data. JAD-filen er en ekstern fil, hovedgrunnen til at den eksisterer er at man skal slippe å laste ned hele JAR-filen for å finne ut hva denne inneholder. JAD-filen er i hovedsak en nedskalert utgave av manifestfilen som befinner seg inne i JAR-filen. Disse beskriver innholdet i JAR-filen; det være seg navnet på programmet i JAR-filen, ikon, hvilken klasse det er som er MIDleten – altså den klassen som tilsvarer main-klassen i vanlig J2SE/J2EE. Utover dette kan den også inneholde andre innstillinger som programmet kan lese underveis – dette kan for eksempel være nyttig ved kompatibilitetsproblemer mellom enheter ved at man kan sette spesielle innstillinger for en gitt enhet i en egen JAD-fil spesielt for den/de enhetene og la programmet lese denne, for så å tilpasse seg underveis i kjøringen av programmet. En JAD og JAR-fil kan inneholde flere forskjellige programmer, og de kalles da en MIDlet-suite.

MIDleten er som nevnt den klassen som tilsvarer main-metoden i J2SE – men den har ingen main-metode. Derimot har den tre forskjellige andre metoder, som blir kalt når programmet starter, når programmet blir stoppet (enten eksternt eller ved at programmet terminerer seg selv) – og en metode som blir kalt når kjøremiljøet vil pause programmet – for eksempel fordi det er en

innkommende samtale. Forøvrig fungerer pakker og klasser stort sett som vanlig.

Det er to forskjellige versjoner av MID-profilen på markedet nå; MIDP 1.0 og MIDP 2.0.

MIDP 1.0 var først ute, og inneholder en svært nedstrippet Java-versjon med støtte for timere, enkel GUI-støtte (2d-opptegning og menyer, tekstbokser og så videre), støtte for persistent lagring (RMS) og nettverk som `HttpConnection`. Dette tillater man å lage enkle programmer med menyer og utseende som ligner på enhetens eget – slik tilfellet med WeGolf er.

MIDP 2.0 er en solid utvidelse av den gamle spesifikasjonen. Ny funksjonalitet er for eksempel støtte for HTTPS-protokollen, støtte for lyd, bedre støtte for GUI-programmering, og funksjonalitet spesielt rettet mot spill (layers, transparente bilder) og sikkerhet.

8.5 Hierarki

Operativsystem -> JVM -> Konfigurasjon -> Profil -> Brukerprogram

8.6 WeGolf og teknologi

WeGolf fokuserer på et kjapt og enkelt program som ikke skal forstyrre brukeren eller være i veien. MIDP 1.0 sine enkle menyer og GUI-funksjonalitet er derfor et godt valg – dette er etterhvert godt innarbeidet, inneholder lite feil og er relativt konsistent fra mobil til mobil, samtidig som det ikke bryter dramatisk med mobiltelefonens eget utseende. I tillegg når WeGolf ut til et mye større marked ved å holde fast på MIDP 1.0 – det er fremdeles svært mange enheter på markedet som ikke støtter den nyere varianten. Ny funksjonalitet som oversiktskart og så videre vil dog antakelig være enklere å implementere på en god måte ved bruk av MIDP 2.0 – og det kan derfor være greit å vurdere å ha to versjoner på markedet – der den nyeste utnytter større deler av mulighetene som de nye mobilene tilbyr. På den måten vil også brukerne få mer igjen for sin investering i dyrt utstyr. Det er mulig at et bedre alternativ dog vil være å ta utgangspunkt i MIDP 1.0 hele veien, men la programmet selv teste for nyere funksjonalitet og utnytte dette om det finnes tilgjengelig. Slik slipper brukeren å vite noe om hvilken profil mobiltelefonen har – svært få brukere vet slike tekniske detaljer.

8.7 Utviklingsmiljø

WeGolf har til nå blitt utviklet i det verktøyet som er regnet som det beste for mobilutvikling; nemlig Netbeans [16]. Netbeans har over lengre tid fokusert sterkt på støtten for mobilutvikling - dette er et område hvor det lenge har vært ganske dårlig verktøystøtte. Netbeans Mobility Pack støtter ved hjelp av GUI-valg hele prosessen fra prosjektopprettelse til utprøving på den fysiske enheten. Den tette integrasjonen mot både mobilen og emulatorer fra de forskjellige telefonleverandørene gjør en ellers ganske kronglete prosess enkel.

Mobiltelefonleverandørene leverer sine egne emulatorer for sine modeller - disse emulatorne kjører ofte den samme programvaren som de faktiske telefonene slik at testing blir realistisk på et stort antall enheter, uten at man faktisk må investere relativt store penger for å få testet mot alle de forskjellige modellene. Det har dog vært et problem at ikke alle emulatorne faktisk gjør alt 100% likt på den fysiske telefonen, for eksempel på grunn av forskjellig programvareversjon, eller rett og slett at emulatoren har kjørt en helt annen kodebase. På litt eldre enheter er dog slike feil stort sett fjernet siden emulatorne relativt lett kan oppgraderes og er en ekstern plugin til Netbeans Mobility Pack.

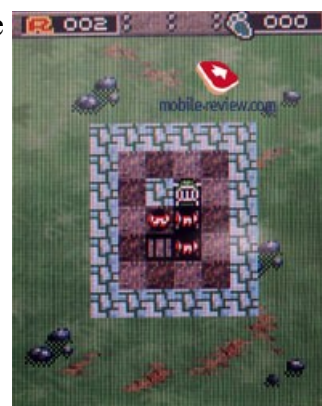
De forskjellige emulatorne gir ofte også mulighet for at man kan velge ytterligere støttede (valgfrie) API - slik at programvaren blir spesialtilpasset for mobiltelefoner med støtte for eksempelvis Messaging, 3D-grafikk og lignende.



Illustrasjon 9: 3D-grafikk i MIDP 2.0

8.8 Grafikkmuligheter

MIDP 2.0 fikk som nevnt utvidet støtten for spillprogrammering. En av de viktigste nye funksjonene som ble lagt til var støtte for flere lag med grafikk som opererer uavhengig av hverandre, og som mobilen selv i større grad håndterer gjennom API. Illustrasjon 2 viser et eksempel på et tile(rute)-basert spill. Her ser vi helt tydelig at det antakelig er brukt såkalte tiles i bakgrunnen, dvs. at grafikken gjerne blir laget i ett stort bilde, og at dette igjen blir delt opp i små biter som settes sammen av de innebygde API'ene i den rekkefølgen programmereren ønsker. Slik sparer man både antall filer, tiden det tar å tegne opp grafikken, og ikke minst slipper utvikleren mye ekstraarbeid med å få programmet til å tegne opp



Illustrasjon 10: Eksempel på tile-basert spill i MIDP 2.0

enkeltpunkter og sette disse sammen til større bilder. Nyere mobiler støtter nå også 3D via egne API; illustrasjon 3 er et eksempel på dette. MIDP 2.0 lar selvfølgelig også utvikleren kombinere disse tile-baserte lagene med andre lag, og ren punkt/strek-tegning over disse igjen. Dette er antakelig noe vi kan dra nytte av i en senere implementasjon av WeGolf, ved at vi tegner opp en mer komplett bane enn det vi i første omgang tar sikte på. På grunn av bruk av gjennomsiktede PNG-bilder har vi i praksis endt opp med å benytte noe MIDP 2.0-funksjonalitet.

9 Samarbeid, prosess og kommunikasjon

9.1 Prosjektplanlegging

Vårt prosjekt har ikke blitt planlagt og designet på den "tradisjonelle" måten med kravspesifikasjon, klassediagram, use cases og lignende. Vi har hatt møter med oppdragsgiver, og deretter har prosjektet "blitt til" underveis. Gruppen har innad hatt forskjellig fokus med noen som fokuserte på koding, mens andre var mest interessert i å lære seg teori. Siden vi hadde såpass frihet i kurset, og at vi egentlig ikke hadde noen skikkelig oppgavebeskrivelse syntes vi at dette fungerte greit. Vi har hatt frie tøyler, og kunne bestemme oss for hva vi ville fokusere på underveis.

Det har gått med en del forskning, både på teori og kodedel. Nye teknologier måtte læres, og det har oppstått enkelte problemer underveis, spesielt i forhold til valg av koordinatsystem på mobiltelefon. Vi satset først på koordinatsystemet som WeGolf benytter, men da avstandsberegning i dette koordinatsystemet (WGS84) krevde en del matematikkfunksjonalitet som ikke er enkelt tilgjengelig bestemte vi oss for å skifte til et annet system (UTM).

Tilgang til artikler har fungert relativt greit, siden vi har hatt en enorm ressurs i Informatikkbiblioteket [29], der man kan søke i flere databaser.

9.2 Samarbeid innad i gruppa

Vi har valgte å ha fast møtetid hver torsdag, med mulighet til å utvide ved behov. Utover i prosjektarbeidet har vi ikke alltid vært like flinke til å holde på den faste møtetiden. I første del av prosjektarbeidet var vi også flinkere til å skrive ned møtereferater og publisere disse for resten av gruppen. I ettertid ser vi at manglende faste møter bidro til at tempoet i prosjektarbeidet ble litt

lavere enn ønskelig i perioder, og at manglende møtereferater bidro til at prosjektarbeidet ble mindre strukturert.

Punkt 4 i dette kapittelet beskriver verktøyene vi har benyttet i arbeidet.

9.3 Kommunikasjon med WeGolf

Kommunikasjonen med WeGolf har stort sett gått over mail, men med noen møter for å informere og diskutere videre arbeid (møte 2. mars og 22. mars).

Etter litt start-problemer har samarbeidet med Werner hos WeGolf gått veldig bra. Vi startet med å sende en mail til mailadressen på nettsiden til WeGolf, men denne havnet hos en annen ansatt i firmaet som ikke hadde hørt om prosjektet og faget vi tok. Etter å ha fått ryddet opp i dette hadde vi noen møter med Werner hvor vi fikk forklart hvordan WeGolf fungerer i dag, hvilke muligheter som finnes og hva han tenkte seg videre. Han var hele tiden veldig opptatt av at vi skulle få gjøre noe vi hadde lyst til og var åpen for prosjektforslag fra oss. Dette gjorde at vi stod fritt til å velge hva vi ville jobbe med, men med et godt produkt som bakgrunn og utgangspunkt for det videre arbeidet.

9.4 Tidsplan

Innlevering undringsdokumenter:	08.02.2006
Tilbakemelding undringsdokumenter:	01.03.2006
Første møte med WeGolf	02.03.2006
Presentasjon av prosjektet i kurset:	22.03.2006
Andre møte med WeGolf	22.03.2006
Presentasjon av artikkel ("Just say Nokia"):	22.03.2006
Innlevering midtveisrapport:	29.03.2006
Tilbakemelding midtveisrapport:	19.04.2006
Innlevering sluttrapport:	10.05.2006

9.5 Verktøy i prosjektet

I prosjektarbeidet har vi tatt i bruk et vidt spekter av forskjellige kommunikasjon- og samarbeidsløsninger. Disse spenner fra asynkrone kunnskapshåndteringsverktøy som Subversion og dokuwiki, via asynkrone kommunikasjonsverktøy som MSN Messenger, e-post og tekstmeldinger

til synkron kommunikasjon over mobiltelefon. Det store utvalget av kommunikasjonsteknologi har gjort oss fleksible i arbeidet; samtidig kan det være problematisk å sørge for at alt som skjer mellom to eller flere mennesker i én kommunikasjonskanal blir formidlet til hele gruppen. Dette krever bevisst innsats fra alle deltakere, og her har vi sett at vi ikke alltid har vært flinke nok til å formidle diskusjoner mellom noen få i én kanal til resten av gruppen via en annen kanal. Et eksempel er ansikt til ansikt-kommunikasjon og lynmeldingkommunikasjon, som i større grad burde vært oversatt til wikiformat, slik at alle fikk innsikt i alle diskusjonene.

Her mener vi absolutt det er behov for en programsuite (gjerne fra open source software-miljøet) hvor e-post, lynmeldinger, tekstmeldinger, versjonskontroll og nettsted er integrert, og hvor informasjon automatisk konverteres mellom de forskjellige mediene.

9.5.1 Integrert utviklingsmiljø - NetBeans 5.0

Til koding har vi benyttet det fritt tilgjengelige NetBeans 5.0[16], som er basert på åpen kildekode. Programmet har god støtte for utvikling av Javaapplikasjoner til mobile enheter, og vi er generelt godt fornøyd med det.

9.5.2 Versjonskontroll - Subversion

SVN[26] står for Subversion, og har spilt en stor rolle i utviklingen av prototypen og dokumentene våre. Målet med Subversion-prosjektet er å lage et versjonshåndteringsystem som er en komplett erstatning av CVS i åpen kildekode-samfunnet. Flere av gruppemedlemene hadde erfaring med SVN fra tidligere kurs og det ble etterhvert bestemt at dette skulle brukes som versjonshåndterings-system.

SVN gjør det mulig å lagre prosjektets kildekode og dokumenter på en server slik at utviklere kan hente det ned til sin lokale maskin. Når en har oppdatert koden kan man enkelt legge til den oppdatere versjonen slik at andre utviklere kan laste den ned og alltid ha siste versjon tilgjengelig.

SVN virker på alle nyere versjoner av Unix, Win 32, BeOS, OS/2 og MacOS X, og på alle filtyper så vi har kunnet ha både javakoden og tekstdokumenter under versjonshåndtering. For å hente koden fra serveren ble TortoiseSVN [27] brukt. Dette er en Subversion-klient implementert som et grafisk interface.

SVN har vært med på å gjøre versjonshåndtering av prosjektet veldig enkelt. Vi hadde alltid full

kontroll over hvem som gjorde hva, og ble det konflikter kunne vi alltid gå tilbake til en tidligere versjon. SVN gjør det enkelt å holde rede på versjoner og faren for å overskriving av dokumenter og misting av data er borte.

På tross av at SVN er ment å brukes på filer som er ren tekst, slik at systemet kan forstå innholdet i filene og ikke bare se på filen som en stor helhet, har vi valgt å bruke SVN til å lagre kopier av rapporten vi jobber med. SVN kan forstå rene tekstfiler, se hva som er forskjellen fra den ene til den andre og forsøke å smelte sammen filer; denne funksjonaliteten mister man når man bruker OpenDocument-formatet. Likevel har vi hatt stor nytte av å bruke SVN til å gjøre den siste revisjonen av rapporten tilgjengelig for hele gruppen, samt å lagre sikkerhetskopier av den.

DokuWiki, som omtales senere i dette avsnittet, tilbyr litt av den samme funksjonaliteten som SVN, i det at wikier holder rede på alle revisjoner av en fil og lar brukeren gå tilbake til en eldre utgave. Imidlertid fungerer DokuWiki utelukkende gjennom nettleseren, og har presentasjon av innholdet på wikien gjennom en webserver som en viktig del av funksjonaliteten i tillegg til versjonskontroll.

9.5.3 Tekstbehandler - OpenOffice.org

Vi har brukt OpenOffice.org [28] til å forfatte rapportene (OpenOffice Writer) og presentasjonen (OpenOffice Impress). OpenOffice er en forholdsvis brukervennlig applikasjon som er helt gratis og tilgjengelig som åpen kildekode. Dessuten har OpenOffice Writer innebygd støtte for PDF-formatet.

9.5.4 Wiki-nettsted for samarbeid - DokuWiki

Vi benyttet oss av en wiki for å lagre informasjon som møtereferater, rapportutkast og diskusjonsoppsummeringer. En wiki er et nettsted hvor eksisterende sider kan redigeres og nye sider kan opprettes meget raskt. Wikien holder rede på historien til en side, slik at man når som helst kan gå tilbake til en tidligere utgave, om man gjør endringer som man angreer på. Brukeren av en wiki trenger ingen kunnskap til HTML; å endre en side er så enkelt som å klikke "Edit" på siden, endre teksten man vil forandre, og trykke "Save". DokuWiki tilbyr enkel syntaks for overskrifter, lister og tabeller.

Wikien har gjort det effektivt og enkelt å gjøre informasjon tilgjengelig for hele gruppen. Informasjon samlet på noen få websider som enkelt kan endres er lettere å holde oversikt over enn for eksempel tilsvarende informasjon spredt utover flere e-poster som ligger i ens innboks. Siden wikisider er så enkle å redigere, kan man omstrukturere, endre og legge til informasjon

kontinuerlig, om man har behov for det. Når en wikiside er opprettet for et tema, vet alle på gruppen at den relevante informasjonen for temaet vil være på wikisiden, og siden wikier er så enkle å redigere er det fort gjort å legge inn informasjon som burde være der, men mangler.

Et problem med DokuWiki er at den ikke kan eksportere innhold til noe annet format enn HTML. Vi hadde likt eksportmuligheter til OpenDocument-formatet som OpenOffice.org benytter, for eksempel. Denne begrensningen har medført at vi har måtte bruke litt tid og krefter på å gjøre om omfattende rapportutkast fra wiki-format til OpenDocument-format.

9.5.5 Lynmeldinger - MSN Messenger

MSN Messenger er et program som lar brukerne "chatte" med hverandre over en internettlinje. Dette har gruppen benyttet seg av når medlemmene har jobbet hver for seg. Det er enkelt å bruke og lett å sende filer til hverandre. Et problematisk aspekt ved denne kommunikasjonsformen er at det ikke uten videre er slik at alle medlemmer i gruppen tar del i samtalen eller får tilgang til loggene fra samtalene i etterkant. I etterkant ser vi at det hadde vært en fordel å være flinkere til å oppsummere noen av diskusjonene vi har hatt over MSN og publisert disse for hele prosjektgruppen på wikien.

9.5.6 E-post

Gruppen har benyttet e-post til å gjøre oppmerksom på nytt innhold på wikien, diskutere møtetidspunkter og andre spørsmål relatert til prosjektet samt kommunisere med WeGolf.

9.5.7 Mobiltelefon

Vi utvekslet på et tidlig tidspunkt mobiltelefonnumre, og disse ble også publisert på wikien. Mobiltelefonen er et nødvendig verktøy når man må få tak i andre gruppemedlemmer på kort varsel og i situasjoner hvor man er på farten.

9.6 Dokumenthistorie og tidligere arbeid

Denne rapporten bygger på Midtveisrapporten for gruppe WeGolf [23], som ble levert 29. mars. Vi har også tatt med noe stoff fra vår presentasjon av artikkelen "Just say Nokia" [24].

9.7 Hva kunne vi gjort annerledes?

Som nevnt tidligere hadde vi en del problemer med å få tak i Werner hos WeGolf i starten av prosjektet. Dette førte til at det gikk noen uker før vi hadde det første møtet med WeGolf og fikk klarhet i hva vi kunne gjøre og hva de ønsket av oss. I etterkant ser vi at vi kanskje burde ha ringt i stedet for å sende e-post for å få en raskere start på prosjektet.

Endelig valg av mål for prosjektet var ikke klart før i slutten av mars. Vi fikk veldig frie tøyler fra WeGolf, og den tiden vi brukte på å finne en konkret problemstilling kunne ha blitt brukt bedre hvis vi tidligere visste hva vi ville eller bare fikk tildelt et prosjekt. Lærdommen vi trekker av dette er at man bør arbeide fokusert for å finne en problemstilling raskt, og om nødvendig be om hjelp til å velge ut et relevant område.

Vi begynte ikke å kode før i påsken grunnet arbeid med midtveisrapporten og mye arbeid i de andre fagene vi tar. Hadde vi vært litt flinkere til å strukturere oss selv og sette opp en mer detaljert framdriftsplan i starten av prosjektet, kunne vi kanskje ha kommet tidligere i gang med kodingen og fått gjort enda mer med programmet. Generelt har vi ikke vært flinke nok til å stille krav til hverandre på gruppen.

Teorien om design av brukergrensesnitt på mobiltelefoner og grafikk på små skjermer som vi har lest har vi i liten grad klart å anvende i prototypen, primært fordi vi kom så sent igang med utviklingen av den.

10 Konklusjon og veien videre

Vi har i løpet av dette prosjektet lært både om golfsporten, teknologi knyttet til utvikling av mobile informasjonssystemer og litt om selve prosjektprosessen.

Vi stod fritt til å velge både prosjekt og fokus innenfor det valgteprosjektet, og vi brukte litt tid på å finne ut hva vi kunne og ville gjøre. Målet vårt var i utgangspunktet å tegne et kart over golfbanen på mobiltelefonens skjerm der punkter på banen som er relevante for golfspilleren er markert. I prototypen har vi implementert visning av hindringene på banen, men uten avstanden til dem. I tillegg vises historikken over tidligere slag som golfspilleren har slått.

Golfsporten hadde 4 av 5 av prosjektgruppas medlemmer liten kjennskap til før prosjektstart, men underveis har vi lært en del om både golfuttrykk og sportens historie, i tillegg til en del om selve

golfspillet.

Om selve prosjektetprosessen har vi fått erfare at det lønner seg å komme tidlig i gang med prosjektet. Vi brukte litt tid på å komme inn i arbeidet, men tok oss inn etterhvert og jobbet bedre etterhvert som problemstillingen ble klarere.

Hadde vi hatt mer tid kunne vi ha laget en prototype hvor vi i større grad fikk anvedt teorien, både fra faget og fra artikler rundt utfordringene små terminaler gir. Vi føler likevel at vi har utvidet vårt begrepsapparat rundt fenomenet mobilitet og fått nyttig konkret systemutviklingserfaring gjennom prosjektet og faget INF5261 Utvikling av mobile informasjonssystemer.

11 Forkortelser

API Application Programming Interface, et grensesnitt som et kodebibliotek, program eller datasystem tilbyr andre programmer, slik at de kan få tilgang til funksjonaliteten som fins i biblioteket / programmet / systemet

GPL General Public License. Programvare publisert under denne lisensen har såkalt "åpen kildekode" slik at hvem som helst kan lese kildekoden og kan fritt benyttes av alle uten noen kostnad, med forbehold om at programmet den benyttes i også må distribueres under GPL.

GPS Global Positioning System. Satellittbasert system for navigasjon.

J2ME Java 2 Micro Edition, også Java Platform, Micro Edition, en samling Java APIer for enheter med begrensede ressurser, for eksempel mobiltelefoner eller vaskemaskiner.

J2SE Java 2 Standard Edition, også Java Platform, Standard Edition, en samling Java APIer tiltenkt bruk på datamaskiner.

UTM Koordinatsystem bestående av kvadratiske ruter som tilsammen utgjøre et rutenett over hele jordkloden.

WGS84 Koordinatsystem som dekker hele jorda med samme rutenettet. Bruker bredde- og lengdegrader.

12 Referanser

1. "The Challenge of Mobile Devices for Human Computer Interaction", Dunlop, Brewster, Personal and Ubiquitous Computing (2002), 6:235-236
2. "Evaluating Interface Design Choices on WAP Phones: Navigation and Selection", Chittaro, Dal Cin, Personal and Ubiquitous Computing (2002), 6:237-244
3. "Rendering effective route maps", M. Agrawala, C. Stolte, ACM SIGGRAPH (2001)
4. "Halo: A technique for visualizing off-screen locations", P. Baudisch, R. Rosenholtz, CHI 2003 (2003)

5. "J2ME Game Programming" - Martin J. Wells (2004) (Thomson Course Technology, ISBN: 1592001181)
6. "Golf Club Technology", Phillip Eide, John Nelson (2001)
http://www.students.dsu.edu/eidep/technology_paper.htm, lest 10.05.06
7. GPS Industries: <http://www.sportandtechnology.com/features/0240.html>, lest 10.05.06
8. "From GIS to LBS: An Intelligent Mobile GIS" – Christopher Frank, David Caduff, Markus Wuersch, Intelligent Spatial Technologies & Dep. Of Geography, University of Zürich
9. WeGolf - <http://wegolf.net>, lest 10.05.2006
10. IntelliGolf - <http://www.intelligolf.com>, lest 10.05.2006
11. mScorecard - <http://www.mscorecard.com>, lest 10.05.2006
12. Mobile Golf Scorer - <http://www.mobilegolfscore.com>, lest 10.05.2006
13. iGolfScorer - <http://www.igolfscore.com>, lest 10.05.2006
14. Wireless18 - <http://www.wireless18.com>, lest 10.05.2006
15. Handago - <http://www.handango.com>, lest 10.05.2006
16. NetBeans - <http://www.netbeans.org>, lest 10.05.2006
17. "Look who's talking", Rheingold (2001) Wired magazine 7-01.
18. "Visualizing the Results of Interactive Queries for Geographic Data on Mobile Devices", Stefano Burigat, Luca Chittaro (2005)
19. "A Visualization Design Repository for Mobile Devices", Volker Paelke, Christian Reimann, Waldemar Rosenbach (2003)
20. "Presenting Route Instructions on Mobile Devices", Christian Kray, Katri Laasko, Christian Elting, Volker Coors (2003)
21. "An Empirical Study of Textual and Graphical Travel Itinerary Visualization using Mobile Phones", Masood Masoodian, Nicholas Lane (2002)
22. "Visualizing Information on Mobile Devices", Luca Chittaro (2006)
http://www.computer.org/portal/site/computer/menuitem.eb7d70008ce52e4b0ef1bd108bcd45f3/index.jsp?&pName=computer_level1&path=computer/homepage/0306&file=cover.xml&xsl=article.xsl

[l&:jsessionid=Gfmtj4pvp1Q1pMZ7yZMm6RJpBvKkPgcYWhGjq7psV4G11HXWQnbj!-428020405](#), lest 10.05.06

23. "Midtveisrapport, Prosjekt WeGolf", INF5261 vår 2006

http://www.uio.no/studier/emner/matnat/ifi/INF5261/v06/prosjekter/WeGolf/Midtveisrapport_endelig_utgave.pdf, lest 8.05.06

24. Presentasjon av artikkelen "Just say Nokia", av WeGolf-gruppen, INF5261 vår 2006

http://www.uio.no/studier/emner/matnat/ifi/INF5261/v06/prosjekter/WeGolf/justsaynokia_presentation.pdf, lest 8.05.06

25. "Just say Nokia", Silberman S (1999). Wired magazine

26. Subversion, versjonshåndteringssystem, <http://subversion.tigris.org>, lest 10.05.06

27. Tortoise SVN, grafisk grensesnitt for Subversion, <http://tortoisesvn.tigris.org>, lest 10.05.06

28. OpenOffice.org: <http://www.openoffice.org>, lest 10.05.06

29. Informatikkbiblioteket, <http://www.ub.uio.no/umn/inf>, lest 10.05.06

30. Google Scholar, <http://scholar.google.com>, lest 10.05.06

31: Wikipedia.org: Om "datum", sitert: 10/05/06

32: Wikipedia.org: Om "UTM", sitert: 10/05/06

33: Wikipedia.org: Om "WGS84", sitert: 10/05/06

34: Geoconv, Javabibliotek for konvertering av geografiske koordinater, <http://www.i3s.unice.fr/~johan/gps/>, lest 10.5.06