“BlueRiders”

Et reaksjonsbasert mobilt nettverksspill der deltakerne samler poeng ved å duellere med andre spillere.

Virkemåte

Spillerne er spredt over et ikke definert geografisk areal, egnet for tett befolkede områder. Når to påmeldte spillere kommer innen Bluetooth sin rekkevidde blir spillerne alarmert om at en annen spiller er i nærheten. Spillerne skal da raskest mulig “trykke på avtrekkeren”, dvs trykke en tast på mobiltelefonen sin. Den raskeste av spillerne får poeng, mens taperen mister poeng. Videre kan spillerne sanke ekstra poeng ved å fysisk kommunisere med hverandre og utveksle “nøkkelordet” sitt. På denne måten kan spillet benyttes til å danne spillallianser (og treffe nye mennesker!).

Regler

1. Geografiske begrensninger.
2. En Konkurranse kan bre seg utover en by eller et helt land, etter ønske.
3. Spillet kan f.eks. starte når et visst antall innen et geografisk område har meldt seg på. (eks. 1000 stk i Oslo)
4. Spillet skal være tidsavgrenset / poengavgrenset
5. Spillet kan strekke seg over et begrenset tidsintervall (f.eks. en uke eller en måned), eller spillet kan avsluttes når en spiller når en gitt poengsum.

Målet med prosjektet

Å utvikle en prototyp som enten demonstrerer at det er mulig å lage spillet, eller som til og med til en viss grad fungerer

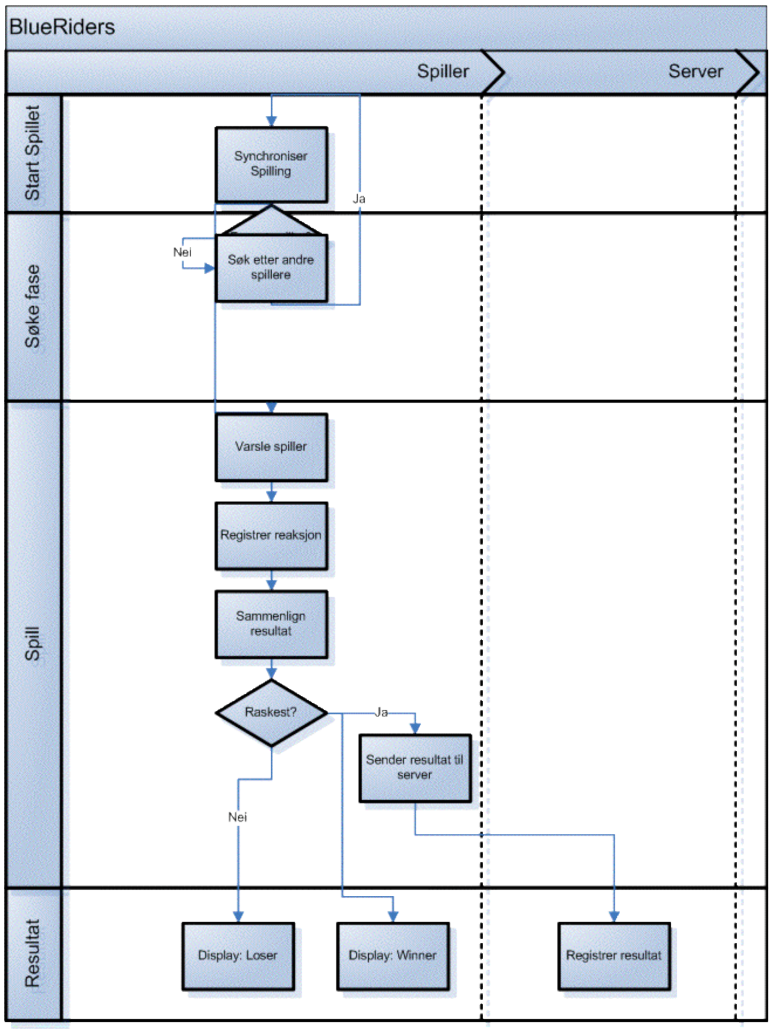
Implementasjon

Grafisk brukergrensesnitt - Spillet skal bygges opp med et enkelt grensesnitt. En lyd/vibrasjon varsler deg når en annen spiller er detektert av mobilen din og en liten melding eller animasjon informerer deg om resultatet av en duell. Utover dette skal spillet være ”usynlig” og kun arbeide i bakgrunnen på mobiltelefonen din. Du kan altså bruke mobiltelefonen din som vanlig. Spillet er teknisk sett et J2ME basert program som utnytter Bluetooth for konstant søke etter andre spillere. I tillegg kan spillet kommunisere med en spillserver via http (f.eks. via WAP og GPRS) eller SMS.

Spillserver - registrerer poeng og rangerer deltakerne i spillet. Serveren skal kunne motta kontaktinformasjon fra mange brukere og samle dette i en database. Videre skal dataen kunne prosesseres og gi nyttig informasjon til spillere og administratorer. Spillserveren kan bruke http for å motta informasjon fra spillerne ved at spillernes mobiltelefoner sender definerte http GET’s. Spillerne kan også sende tekstmeldinger til spillserveren. Spillserveren kan også sende ut tekstmeldinger til spillerne, f.eks. når et nytt spill begynner.

Status for prosjektet av 1. April

Teknisk problem	Status	Kommentar
Bluetooth søk / kommunikasjon	Under utarbeiding	Lite utprøvd standard Eksempler utilgjengelig
Synkronisering av Bluetooth	Venter	Trenger punkt 1
Opprettholde ustabil forbindelse	Venter	Trenger punkt 1
Kjøre J2ME i bakgrunnen	Nedprioriteter	Ikke viktig i prototype
Duell for flere enn 2 spillere	Avklart	Flagg som viser aktivitet ved flere spillere
Gjenntagende like spillere	Avklart	Tidsramme for kombo-spill
kontakt for server-registering	Ferdig	
Databasemodell for server	Ferdig	Se videre i dokumentet
Protokoll mellom spill og server	Ferdig	Se videre i dokumentet
Implementering av Server	Venter	
Lage visuellt grensesnitt	Venter	Uviktig inntil videre
Web-grensesnitt	Venter	Uviktig inntil videre
Utprøvning av Prototype	Venter	
Analyse av eksperiment	Venter	



En normal spilling

En liten applikasjon skal kjøre I bakgrunnen, som søker etter andre Bluetooth-enheter som kjører det same spillet. Hvis en annen enhet som kjører det samme programmet kommer inn i kommunikasjons-radiusen for Bluetooth-kommunikasjon, vil de to initsiere kontakt, og si ifra til brukerene. Etter det må brukeren prøve å trykke på en knapp på mobiltelefonen før den ande brukeren. Den første personen som trykker vinner og den andre taper. Vinneren får poeng lagret på serveren og taperen mister poeng. Bonuspoeng kan utveksles hvis spillerene tar kontakt i virkeligheten.

BlueRiders protokoll mellom spillserver og klient

BlueRiders protokollen er en protokoll som er bygget over en allerede eksisterende transportprotokoll. Selv om BluRiders-protokollen er transparent for hvilken unnerliggende protokoll som brukes, ser vi det hensiktsmessig å bruke http og/eller

SMS (tekstmeldinger) som underliggende transportprotokoll.

Dersom BlueRiders-protokollen brukes over http, kan ikke spillserveren ta initiativ til en kommunikasjon. Det er kun klienten som kan kontakte spillserveren over http, og spillserveren kan kun svare på en forespørsel fra klienten. Dersom SMS brukes kan både spillserver og klient starte en kommunikasjonsprosess.

Bruk av BlueRiders protokollen over http:

Spillserveren til BlueRiders har domenenavnet blueriders.rute.net. Klienten kan sende kommandoer til spillserveren ved å gjøre en http GET med følgende struktur:

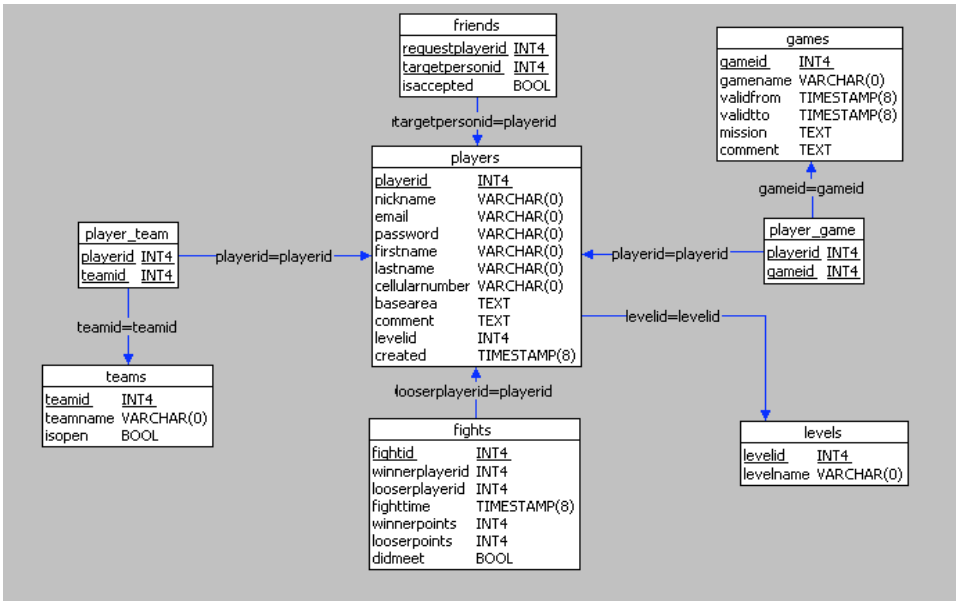
http://blueriders.rute.net/?p=<personid>&c=<command>[&<param>=<value>]*

Alle kommandoer må inneholde personid'en til spilleren på klienten og en kommando. En kommando kan i tillegg ha ekstra obligatoriske parametere.

Kommando	Parameter	Mulige svar fra serveren	Kommetar
Registerfight	l=<looserplayerid>	ok illegallooserid missingobligparameter	Vinner av en duell sender denne kommandoen for å registrere duellen
Registerfriend	f=<friendplayerid>	ok illegalfriendid missingobligparameter	En spiller som ønsker å foreslå et vennskap med en annen spiller, sender denne kommandoen

Kommentarer til databasen

Databasemodell



Databasen er implementert i PostgreSQL 7.4 og kan aksesseres med følgende info:

Host: alpha.dakantus.no
Port: 5432 (default for PostgreSQL)
Username: blueriders
Password: ********

Forklaring til databasemodellen

Her følger en forklaring av alle tabellene i databasemodellen. Forklaringen definerer hvordan de forskjellige tabellene og attributtene skal brukes.

players

Alle deltakerne i BlueRiders må være registrert i denne tabellen.

playerid:	En unik id for hver spiller. Denne verdien er skjult for spilleren.
nickname:	Hver spiller har et unikt nickname som spilleren kan endre etter ønske.
email:	Epostadressen til spilleren.
password:	Spilleren oppgir et passord. Dette brukes bl.a. når spilleren logger seg inn på hjemmesidene til BlueRiders for å administrere kontoen sin der.
firstname:	Fornavnet til spilleren
lastname:	Etternavnet til spilleren.
cellularnumber:	Mobilnummeret til spilleren. Dette brukes bl.a. av serveren når serveren ønsker å sende en SMS til spilleren.
basearea:	Dette er en beskrivelse spilleren skriver over hvor vedkommende ”vanker”. Tanken er at andre spillere kan oppsøke dette området for en duell.
comment:	Dette attributtet har ikke et strengt definert innhold, og kan brukes til hva som helst. Spilleren selv kan verken se eller editere dette attributtet.

levelid: Forteller hvilket nivå en spiller er på. En spiller kan stige i nivå ved å oppnå forskjellige kriterier. Kriterier kan f.eks. være en kombinasjon av hvor lenge spilleren har vært med i spillet og hvor mange dueller spilleren har vært med i (eller vunnet).
created: Inneholder datoen spilleren registrerte seg i spillet. Dette kan brukes i utregning av nivået til spilleren.

levels

Denne tabellen inneholder alle mulige nivåer en spiller kan være på. Alle spillerne i BlueRiders er på ett av disse nivåene. En spiller kan f.eks. opparbeide seg et høyere nivå ved å være en aktiv spiller (være med i mange dueller). Tanken er at det skal gi status å ha et høyt nivå. Samtidig vil andre spillere få flere poeng dersom de vinner en duell mot en spiller med et høyt nivå.
levelid: hvert nivå har en unik livelid.
levelname: hvert nivå har et navn.

friends

Spiller kan definere hvem som er venner. Dersom en spiller kommer innenfor rekkevidde til en venn, varsler ikke telefonen en duell, men kan kanskje heller informere om at en venn er innenfor rekkevidde. Dersom du bor sammen med eller oppholder deg mye sammen med en annen person som også er en BlueRider, er det irriterende om mobilen din varsler duell med denne personen hele tiden, og man kan derfor definere denne personen som en venn. Begge parter i et vennskap må akseptere vennskapet.
requestplayerid: inneholder playerid til spilleren som foreslår et vennskap. (initiativtaker)
targetplayerid: inneholder playerid til spilleren som initiativtaker ønsker et vennskap med.
isaccepted: en boolean som forteller hvor vidt offeret har akseptert vennskapet eller ikke. Et vennskap er kun gyldig dersom isaccepted=true.

teams

Det er mulig å lage klaner. En klan er en gruppe av spillere som samarbeider. Dersom en spiller kommer innenfor rekkevidde til en annen spiller i samme klan, varsles ikke spillerne om duell, men kan bli informert at et klanmedlem er innenfor rekkevidde. En klan har en felles poengsum. Poengsummen til en klan er summen av poengsummen til alle spillerne som er med i klanen. BlueRider- spillet vil ha en poengsumliste for klaner og en annen for enkeltspillere. Alle spillere er med i poengsumlisten for enkeltspillere, og poengene for en spiller som er med i en (eller flere) klan(er) teller også i poengsumlisten over klanene.
teamid: En unik id hver hver klan
teamname: Navnet på klanen
isopen: En boolean som forteller om klanen er åpen. Dersom en klan er åpen kan hvem som helst bli med i klanen uten videre. Dersom klanen ikke er åpen må en av klanens medlemmer akseptere en spiller som ønsker å bli med i klanen.

player_team

Denne tabellen linker en spiller til en klan. På denne måten kan en spiller være med i flere klaner, og en klan kan ha flere spillere.
teamid: Inneholder en teamid.
playerid: Inneholder en playerid.

fights

Her lagres alle dueller mellom spillere. En duell er alltid mellom to spillere, og en duell registreres med kun en tuppell i denne tabellen.

fightid:	En unik id for hver duell
winnerplayerid:	playerid til vinneren av denne duellen.
looserplayerid:	playerid til taperen av denne duellen.
fighttime:	Tidspunkt for duellen.
winnerpoints:	Hvor mange poeng vinneren av denne duellen fikk. Dette kan varierer etter hvilket nivå vinneren og taperen hadde.
looserpoints:	hvor mange poeng taperen av denne duellen fikk. Dette kan varierer etter hvilket nivå vinneren og taperen hadde.
didmeet:	Dette er en boolean som forteller om de to duellerende spillerne møttes etter duellen. Spillerne kan få en ekstra bonus dersom de møtes etter en duell.

games

Det kan kjøres flere separate spill i BlueRiders. Tanken er at det alltid kun er ett aktivt spill. Derfor kobles ikke fights til games, da dette er gitt av tidsrommet et spill strekker seg over og fighttime i fights. En spiller må melde seg på et spill for å være med i spillet. Alle spill går over et definert tidsrom, og har et mål. Sluttidspunktet for et spill kan enten være en definert dato, eller når noen har klart målet til spillet. Dersom det siste er tilfellet, blir sluttdatoen til spillet først definert når spillet er avsluttet.

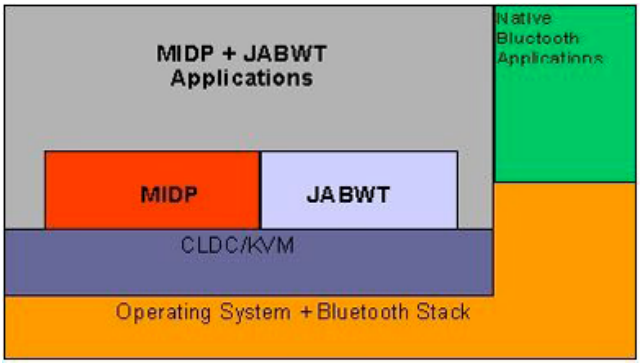
gameid:	En unik id for hvert spill.
gamename:	Navnet til spillet.
validfrom:	Spillets startsdato. Spillet kan starte en gitt dato, eller når bestemte kriterier er oppnådd (f.eks. når 1000 spillere er påmeldte). Dersom det siste er tilfellet skal valdifrom=NULL inntill startdato er definert.
validto:	Spillets sluttdato. Spillet kan slutte en gitt dato, eller når bestemte kriterier er oppnådd (f.eks. når en spiller har oppnådd en gitt poengsum eller klart målet). Dersom det siste er tilfellet, skal validto=NULL inntill sluttdato er definert.
mission:	Dett er en tekst som beskriver målet med spillet eller oppdraget. Dette kan være å oppnå flet poeng, vinne en duell mot en bestemt person eller noe helt annet. Denne teksten leses av spillerne når de melder seg på et spill.
comment:	Dette er et attributt som er usynlig for spillerne. Innholdet i dette attributtet er ikke strengt definert, og kan inneholder administrative kommentarer til spillet.

player_game

Denne tabellen linker spillere til spill. På den måten kan en spiller være med i flere spill og et spill kan ha flere spillere.

playerid:	playerid til en spiller som er med i et spill.
gameid:	gameid til spillet spilleren er med i.

Første Prototype



For å kunne skrive blutooth applikasjoner i java er det behov for et bluetooth java API. Dette API'et har blitt definert av en ekspergruppe representert av 21 selskaper. Arbeidet med denne starte desember 2000 under "The Java Community Process". Arbeidet med å utvikle bluetooth API'er ble gitt betegnelsen Java Specification Request 82, eller JSR-82.
Kontaktopprettelse mellom to enheter må verifiseres og autentiseres, før to enheter kan utvelske informasjon. Dette gjør at den planlagte metoden for å oppdage andre enheter og initiere spill kan bli hindret av protokollenen for å opprette en kontakt.En alternativ mulighet for oppdagelse av deltaker kan være:
Dersom alle deltakere lar sitt telefon navn være "blue+mobilnr", f.eks blue41295402 kan en annen deltaker finne ut følgende i det øyeblikk en deltagende enhet er funnet:

1. De fire første bokstavene i navnet indikerer at enheten er med i spillet
2. Siden enheten er med i spillet vil karakter 5 t.o.m 12 inneholde deltakerens mobilnr.

Dersom det er mulig å hente navnet fra "server" (connection.getNaturalName())
uten å gå gjennom "services" oppdagelse kan vi slippe billig unna på følgende vis:
Når en spillet(applikasjonen) oppdager en motstander varsler det spilleren. Spilleren kan da ved hjelp av å trykke på "skyt" sende beskjed til serveren at deltaker med nummeret som spillet(applikasjonen) har funnet, er skutt. Serveren kan da sende beskjed videre til den som er skutt. En helt klar fordel vi får ved å gjøre det på denne måten er at det kreves meget liten tid der mobiltelefonene trenger å være i nærheten av hverandre for at spillet skal gå sin gang. Hvis spillerne sitter i hver sin bil som kjører motsatt vei kan man skyte eller bli skutt selv om man 5 sekunder etter oppdagelse har skjedd allikevel skyte eller bli skutt. Det kan også hende at bare den ene rekker å se den andre.

Plan:

Kjetil = Kj
Kim = Ki
Eivind = E

Oppgave	Ansvarlig	Frist
Levering av undringsdokument	Kj, Ki, E	12.02
Lage applikasjon som gjør at to Bluetooth telefoner gir melding om at de har oppdaget hverandre. Programmet skal gi et lydsignal	Kj, Ki	18.03
Definere protokoll mellom spill og server	E	18.03
Definere databasemodell for server	E	18.03
Synkronisering av oppdagelses signal. Definere toleransegrense	Kj, Ki	01.04
Implementere databasemodell og serversiden av protokollen. Lage et enkelt web grensesnitt til bruk for testing av kommunikasjonen mellom mobiltelefon og server. Installere systemet på en server tilgjengelig via Internet.	E	01.04
Sende “oppdaget” melding til server		01.04
Oppdatert prosjektdokument	Kj, Ki, E	01.04
Lage visuelt grensesnitt. Påskenøtt	Kj, Ki	15.04
Lage web grensesnitt for spillere: Highscore, opprettelse av gjenger (lag)	E	15.04
Utprøvning av eksperimentell applikasjon	Kj, Ki, E	01.05
Analysere eksperimentet, trekke mulige konklusjoner med hensyn på relevante spørsmål.	Kj, Ki, E	
Innlevering av rapport	Kj, Ki, E	13.05