

Positiebepaling met WiFi-signalen

Groep H3

Probleemoplossen en ontwerpen, Departement Computerwetenschappen

22 mei 2007

Overzicht

- 1 Inleiding
- 2 Omschrijving van de applicatie
- 3 Implementatie
- 4 Algoritme voor plaatsbepaling
- 5 Gebruikersinterface
- 6 Proces
- 7 Besluit

Groepsleden

- Cedric Cuypers
- Chris Adriaensen
- Daniel Balog
- Guy Van Den Broeck
- Jef Maerien
- Philippe Dellaert
- Wouter Van Ranst

Groepscoördinator: Philippe Dellaert

Secretaris: Chris Adriaensen

Omschrijving van de applicatie

Sheephunt



Concept

Een spel waarbij verschillende spelers zo snel mogelijk een aantal schapen proberen te vangen en daarbij gebruik kunnen maken van bommen en powerups.

- Plaats: Elke locatie met voldoende WiFi-stations
- Doelpubliek: Kinderen ouder dan 7 jaar en volwassenen

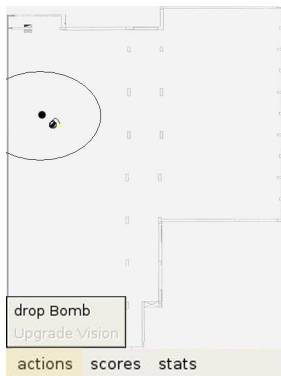
Spelverloop

- De speler loopt door het gebouw of op de locatie, op zoek naar een schaap.
- Hij/zij kan:
 - een schaap vangen, waarna er een nieuw verschijnt
 - een schatkist oprapen
 - een bom plaatsen
 - een vision powerup gebruiken
 - over een bom van een tegenspeler lopen

Zichtbaarheid

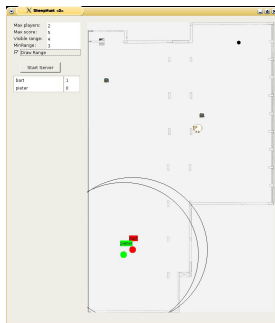
Verschillende zichtbare voorwerpen:

- Schaap
- Schatkist, bevat een bom of een vision powerup
- Eigen geplaatste bom



Server

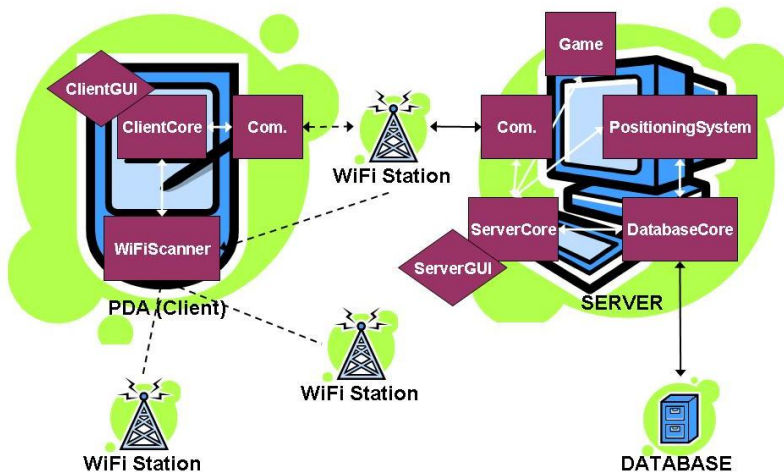
- Instellen van:
 - Aantal spelers
 - Aantal te vangen schapen
 - Zichtbaarheid
 - Afstand nodig om iets op te rapen of te vangen
- Server starten met de instellingen
- Spelverloop volgen



Implementatie

- 3 Implementatie
 - Verdeling
 - Client
 - Communicatie
 - Server

Onderdelen



Client

ClientCore

Wordt op de PDA gestart en start de GUI en de updater.

Updater

Gebruikt een aparte thread, doet WiFi-metingen en stuurt deze samen door met eventuele gebruikeracties door naar de server.

UserGui

Toont de GUI en vangt events op waarbij de gepaste actie wordt uitgevoerd.

Communicatie

- Afgestapt van RMI
- Eerst XML-RPC geïmplementeerd
- Later overgeschakeld op sockets ontwerp

Server

ServerGui

Start de server en toont de GUI.

ServerCore

Maakt een DatabaseCore object, een PositionCalculator object en een Game object aan.

Game

De logica en ai van het spel.

DatabaseCore

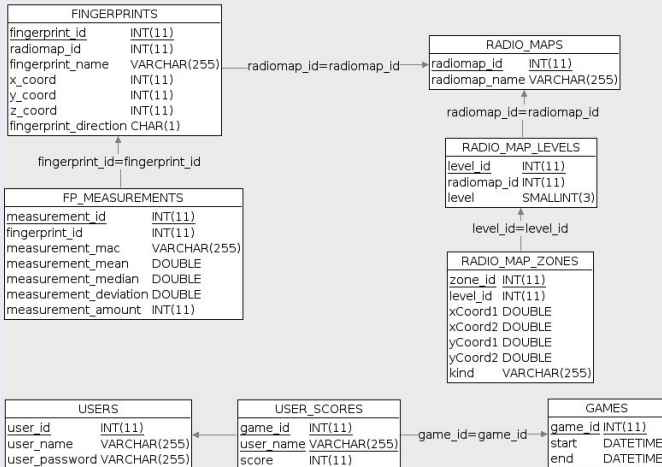
De laag tussen de server en de database (MySQL met JDBC).

PositionCalculator

Is verantwoordelijk voor de positiebepaling.

Database

Database design



Algoritme

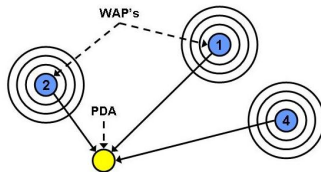
4

Algoritme voor plaatsbepaling

- Fingerprinting methode
- Structuur
- K-nearest-neighbours methode
- P-previous-k-nearest-neighbours methode
- Particle filer
- Experimenten

Fingerprinting methode

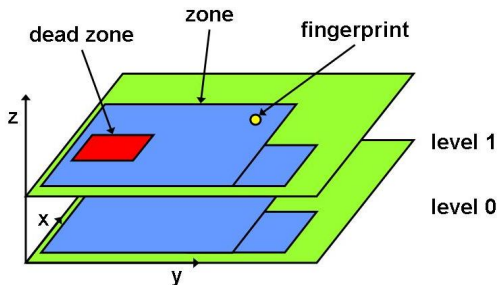
- Opmeten van de ontvangen signaal-sterkten van de verschillende WiFi-stations op een aantal posities op de locatie
- Positie van nieuwe meting benaderen door de 'dichtsbijzijnde' fingerprint
- Verschillende methodes om de 'dichtsbijzijnde' fingerprint te berekenen
- Experimenten \Rightarrow Beste: niet-lineaire exponentiële 2 en de ∞ -norm



Structuur

Per locatie één of meerdere Radiomaps met de volgende informatie:

- levels
- zones per level
- deadzones per zone
- fingerprints



K-nearest-neighbours methode

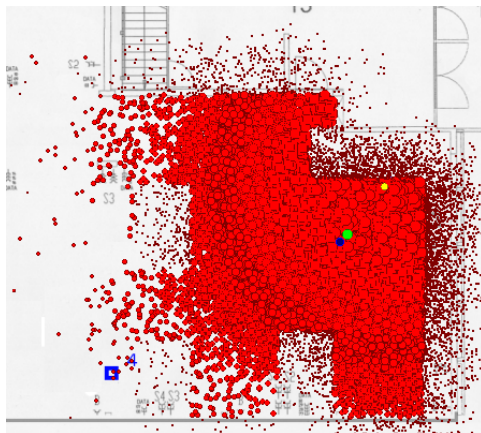
- Uitbreiding op de fingerprint methode
- Niet één, maar meerdere fingerprints
- Gewogen midden tussen deze fingerprints
- Parameters bepalen de gewichten van elke fingerprint

P-previous-k-nearest-neighbours methode

- Uitbreiding op de k-nearest-neighbours methode
- Rekening houden met vorige posities
- Parameters bepalen hoeveel er rekening gehouden wordt met vorige posities

Particle filter

- Stelt een discretisatie van de kansdichtheid van de positie op
- Verschillende fases



Particle

- Gemodelleerde speler die realistisch beweegt
- Heeft een gewicht dat de waarschijnlijkheid aanduidt van de plaats van de echte speler
- Positie aangepast tijdens de predictie-fase
- Gewicht aangepast tijdens de update-fase

Algoritme

Er worden verschillende fases doorlopen:

- 1 Resampling-fase
- 2 Predictie-fase
- 3 Update-fase
- 4 Normalisatie-fase
- 5 Bepalen van de positie

Algoritme - Resampling

- Eliminatie van particles met te kleine gewichten
- Systematic resampling
- Niet elke iteratie
- Experimenten \Rightarrow best resamplen als *Effective Sample Size* (ESS) 50% bedraagt

Berekenen ESS

$$cv_t^2 = \frac{\text{var}(w_t(i))}{E^2(w_t(i))} = \frac{1}{M} \sum_{i=1}^M (Mw_t(i) - 1)^2$$
$$ESS_t = \frac{M}{1 + cv_t^2}$$

Algoritme - Predictie

Realistisch voorspellen van de beweging van de particles.

- Aanpassen versnellingsvector particles
- Traag of snel bewegende particles: willekeurige versnelling
- Gemiddelde snelheid: kleine versnelling
- Altijd voorkeur voor afremmen
- Tijd tussen metingen in rekening brengen

Algoritme - Update

Gewichten van de particles realistisch aanpassen.

- Plaats bepaald via p-previous-k-nearest-neighbours
- Hier rond een normaalverdeling:

$$P(x^{k+1}|z) = \frac{1}{\sqrt{2\pi}\sigma_d} e^{\frac{-d_l^2}{2\sigma_d^2}}$$

- σ_d experimenteel bepaald op 2,5 roosterpunten
- Particles buiten de map of in deadzone \Rightarrow gewicht nul
- Correcte particles \Rightarrow gewicht vermenigvuldigd met berekende waarschijnlijkheid

Algoritme - Normalisatie

- Delen van alle gewichten door hun som
- Vermijden van overloop en onderloop

Algoritme - Positie bepalen

Bepalen positie uit 20000 particles, verschillende methodes:

- Beste particle

$$\Rightarrow X_s = X_s^{max}$$

- Gewogen gemiddelde

$$\Rightarrow X_s = \sum_{i=1}^M (X_{s,i} W_i)$$

- Robuust gemiddelde ($\varepsilon = 2,4$ rasterpunten, blijkt optimaal)

$$X_s = \sum_{i=1}^M (X_{s,i} W_i) : |X_{s,i} - X_s^{max}| \leq \varepsilon$$

Ideale experimenten

- P-previous-k-nearest-neighbours
⇒ Gemiddelde fout: 2,2 meter
- Particle filter
⇒ Gemiddelde fout: 70cm
- Duur: 4 ms

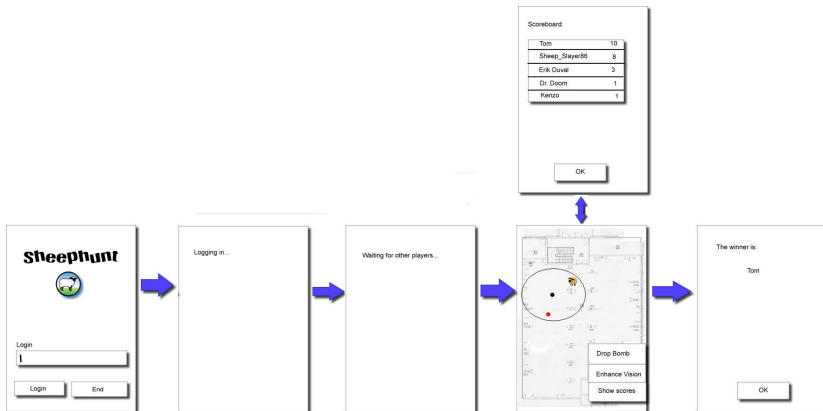
Realistische experimenten

- Via eigen ontwikkelde *walktool*
- Heel veel experimenten/walks
 - 20 keer 256 combinaties van gewichten
 - Verschillende bewegings modellen
 - Aantal particles
 - *Effective Sample Size*
 - ...
- Gemiddelde fout:
 - Stilstaan tot rustig bewegen: 2,5 meter
 - Sneller bewegen: 3,5 - 4,5 meter
- Duur (20000 particles): 50 ms

Gebruikersinterface

- 5 Gebruikersinterface
 - Onderzoek
 - Interface

Eerste ontwerp

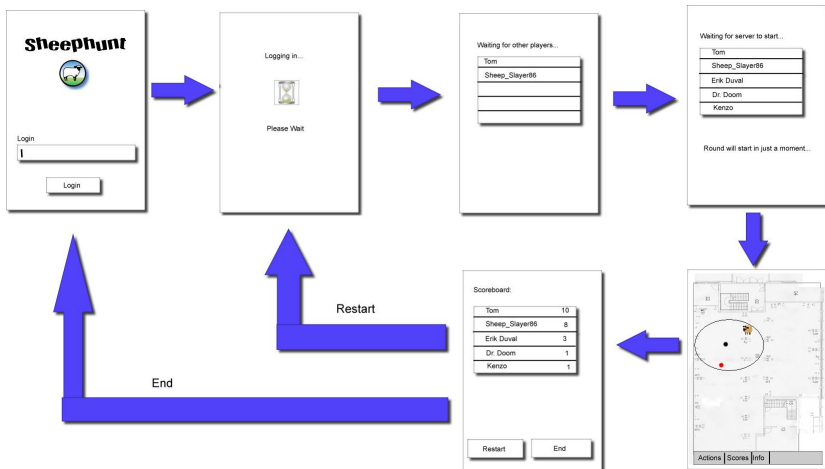


Onderzoek bij het doelpubliek

Problemen:

- Knoppen
- Tijdens het wachten niet weten wie ingelogd is
- Apart score scherm
- Welke items?
- Alleen de winnaar zichtbaar op het einde
- END-knop

Standaard gebruik



Login scherm

Sheephunt



Login

Login

Lobby scherm

Waiting for server to start...

Tom

Sheep_Slayer86

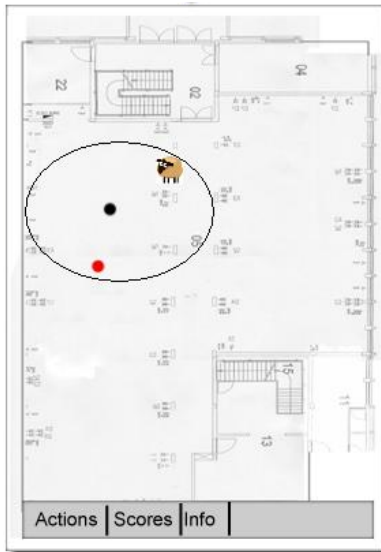
Erik Duval

Dr. Doom

Kenzo

Round will start in just a moment...

Spel scherm



Eind scherm

Scoreboard:

Tom	10
Sheep_Slayer86	8
Erik Duval	3
Dr. Doom	1
Kenzo	1

Restart

End

Proces

6 Proces

- Welke moeilijkheden hebben we ondervonden?
- Wat hebben we bijgeleerd?
- Hoe was de ervaring bij het werken in groep?

Moeilijkheden die we ondervonden hebben:

- PDA
 - Batterij
 - WiFi
 - Soms te weinig
- WiFi-verbinding

Wat we hebben bijgeleerd:

- Java
- XML-RPC/Sockets
- GUI's/SWT
- Plaatsbepalings algoritmes
- Testen
- Teamwerk

Ervaringen bij het werken in groep

Positief

- Motivatie
- Sfeer
- Communicatie
- Opdelen van verantwoordelijkheden
- Flexibiliteit

Negatief

- Wachten

Besluit

- Alles werkt
- Goeie groep
- Interessant onderwerp
- Veel bijgeleerd

Graag willen wij de volgende personen nog bedanken

- Prof. Dr. Ir. Philip Dutré
- Prof. Dr. Ir. Erik Duval
- Prof. Dr. Ir. Dirk Roose
- Prof. Dr. Ir. Marc Van Barel
- Bert Vanhooft
- Michael Meire, Peter Rigole, Ward Van Aerschot, Joris Vanbiervliet, Bart Vandereycken