

AS-84.3147 Automaation käyttöliittymät

# Harjoitustyön loppuraportti

Ryhmä 4

Backman, Hacklin, Käki, Pohjoranta

11.5.2007

# Karhukoplarobotin käyttöliittymä-suunnitelma (eli mitä siitä piti tulla, 29.3.2007)

*"Hyvin tarkkojen suunnitelmien tekeminen on usein täysin turhaa, sillä hyvin harvoin mitkään, väljätkään, suunnitelmat sellaisenaan toteutuvat. Toimimista itse toiminnan hetkellä suunnitelma kuitenkin helpottaa ja siksi kannattaa aina olla jokin suunnitelma - etenkin varasuunnitelma."*

-- Tuntematon varusmieskouluttaja

## 1 Käyttöliittymätoteutuksen lähtökohdat ja tavoitteet lyhyesti

### Lähtökohdat

Karhukoplarobotti eli toteutettavalla käyttöliittymällä ohjattava laite sekä sillä mahdollisesti toteutettavat toiminnot ovat ennalta annetut, eikä niitä muuteta tämän harjoitustyön myötä. Robotilla ja sen käyttöliittymän avulla toteutettavat tehtävät on määritelty väljästi tehtävänannossa: *"Robotilla on tarkoitus etsiä uhat sokkelomaisesta sisätilasta ja neutraloida ne. Käyttöliittymän pitäisi mahdollistaa mahdollisimman tehokas toiminta tehtävän aikana."*

Tyypillinen Karhukoplarobotin (tuttavallisemmin *Kakoron*) käyttäjä on tehtävään koulutettu sotilas-, poliisi- tai vartijatyöntekijä. Käyttäjän tehtävänä ja perimmäisenä mielenkiinnon kohteena on työn kohdetilan sekä siellä olevien uhkien (tai muiden mielenkiinnon kohteiden) henkilökohtainen hahmottaminen ja kartoitus sekä näiden neutralointi. Kakoro on käyttäjälleen ennen kaikkea tärkeä apuväline - ei ainoa työkalu tai mahdollisuus tehtävän toteuttamiseen. Usein Kakoron käyttötilanne on luonteeltaan jännittynyt, mahdollisesti kiireinen ja virheet robotin käytössä saattavat kostautua kalliisti. Kakoron käyttöympäristö on toistaiseksi rajattu valaistuihin, rakenteellisesti selväpiirteisiin sisätiloihin, joissa ei ole robotin havainto- tai kulkuvälineiden toimintaa haittaavia elementtejä kuten lasiseiniä, savua, portaikkoja tai muuta vastaavaa.

Toistaiseksi käyttöliittymän suunnittelussa ei mietitä kaupallisia tekijöitä, kuten kilpailevia tuotteita tai käytettävissä olevia rahallisia resursseja.

### Tavoitteet

Tavoitteena on toteuttaa käyttöliittymä, joka mahdollistaa robotin nopean, virheettömän ja -vaikkei tämä välttämätöntä olekaan - intuitiivisten toimintojen perusteella tapahtuvan ohjauksen, niin normaalikäytössä kuin hätätilanteessa. Ohjaus tarkoittaa paitsi itse robotin ajamista sekä robotissa olevien havaintovälineiden käyttöä, myös robotin käyttöönottoon ja sammuttamiseen liittyviä toimenpiteitä. Mikäli robottiin kuuluu jokin mahdollisten uhkien eliminointiin tarkoitettu työväline, sisältyy myös tämän käyttö robotin ohjaukseen. Lisäksi tavoitteena on, että käyttöliittymä mahdollistaa robotin käyttäjälle robotin tehtävän kohteena olevan tilan hahmottamisen visuaalisesti, sekä tilan pohjapiirroksen ja siihen sijoittuvien tärkeiden kohteiden hahmottamisen graafisesti kartan muodossa.

## 2 Käyttöliittymäsuunnitelman eri tasot

### Käyttöliittymälaitteisto

Käyttöliittymä on kaksisuuntainen. Viestintä robotilta käyttäjän suuntaan tapahtuu (toistaiseksi) yksinomaan tietokoneen ruudulle piirrettävän käyttöliittymän avulla. (On mahdollista, että käyttöliittymä välittää myös esim. Kakoroon asennettuun mikrofoniin tulevat äänet käyttäjälle, jolloin käyttöliittymälaitteistoon on liitettävä jonkinlainen kaiutin.) Viestintä käyttäjältä Kakorolle päin tapahtuu tavallisen tietokonehiiren ja näppäimistön avulla.

### Käyttäjälle tulevan informaation taso - mitä käyttäjälle näkyy

Kakoron käytössä on kolme tilaa; käynnistystila, käyttötila ja sammutustila. Kussakin käyttöliittymänäkymä on erilainen.

Käynnistystilassa käyttöliittymä ilmoittaa käyttäjälle, että robottia käynnistetään (viestiyhteyttä luodaan, kameraa kalibroidaan etc.) ja ilmoittaa onnistuneesta tai epäonnistuneesta käynnistyksestä. Vastaavat tiedot näkyvät käyttäjälle myös sammutustilassa, mutta sammutuksen osalta. Toistaiseksi ei epäonnistuneen käynnistykseen tai sammutukseen tapahtuessa käyttöliittymän antamia ohjeita suunnitella. Tyypillisesti käynnistys ja sammutus ovat nopeita, joten käyttöliittymä ei myöskään viivy pitkään käynnistys- ja sammutustilassa.

Käyttötila on niin robotin kuin käyttöliittymän tavallisin tila. Käyttötilassa käyttäjälle näkyvät (lopullisessa käyttöliittymässä toteutuneet toiminnot on merkitty vihreällä värillä):

- robotin videokameran kuvaama kuva sellaisenaan
- robotin havainto- ja kulkulaitteista saatavien mittausten perusteella muodostettu kartta kohdetilasta
- viesti-ikkuna, johon tulostuvat mahdolliset virhe- tai muut ilmoitukset
- tilailmaisimet robotin tilasta (akun tila, vauriot)
- muut ilmaisimet esim. numeerisesta datasta (robotin nopeus, suuntakulma etc.)
- tilailmaisimet käyttöliittymän tilasta (mitkä näkymät tai optiot ovat aktiivisina)
- graafiset käyttöpainikkeet (hätäseis, Help-painike, etc.)
- ilmoitukset tai muu ”viesti” käyttäjän toiminnoista ja niiden vaikutuksista

Lisäksi videokuva- ja karttakuvanäytöissä näkyvät:

- robotin paikka ja kulkusuunta (kartassa)
- robotin videokameran osoittama suunta (kartassa)
- kohdetilasta jo kartoitettu alue esteineen, karttaan merkittyine kohteineen sekä kohdealueesta vielä tuntematon alue (kartassa)
- uusimman mittausdatan perusteella kartoitettu alue (kartassa)
- laseretäisyysmittarin mittauskeila (video- tai karttakuvassa)
- konenäön avulla tunnistetut uhkakohteet (videokuvassa)
- mahdollisesti robotin ”eliminointityökalun” osoittama kohde tai suunta (sekä kartassa että videokuvassa)
- etäisyys videokuvan keskipisteessä näkyvään, tilan lattialla olevaan kohteeseen kun kamera osoittaa vaakatason alapuolelle (videokuva)

Käyttötilassa käyttäjä voi myös halutessaan kytkeä joko karttakuvanäytön tai videokuvanäytön koko tietokoneen ruudun kokoiseksi.

Informaation visualisointi ja käyttäjälle näytettävien ilmaisimien ynnä muiden toteutus (eli miten kaikki näytetään) noudattaa esimerkiksi normaalissa PC-käytössä hyviksi havaittuja yksinkertaisia väri- ja muita kuvakoodeja.

### Toimintotaso - mitä käyttäjä voi tehdä

Käynnistys ja sammutustilassa käyttäjä ei voi tehdä Kakorolla tai käyttöliittymällä yhtään mitään. Käyttöliittymä siirtyy käynnistystilaan kun järjestelmä kytketään päälle ja käyttöliittymä siirtyy käyttötilaan automaattisesti onnistuneen käynnistyksen jälkeen.

Käyttötilassa käyttäjän tulee pystyä suorittamaan satunnaisessa järjestyksessä ja mielivaltaisen määrä peräkkäin seuraavia *robotin ohjaukseen liittyviä* toimintoja:

- ajamaan robottia eteen, taakse ja sivulle vapaasti valitsemansa määrää (laitteiston mahdollistamissa rajoissa)
- ajamaan robotti johonkin käyttöliittymän kartalta osoittamaansa pisteeseen (tilan asettamissa rajoissa)
- kääntämään robotin videokameraa sivulta sivulle vapaasti
- kääntämään robotin videokamera helposti nolla-asentoon (suoraan eteen) sekä kohdistamaan videokamera kohti jotain videokuvassa näkyvää kohdetta
- kääntämään robotti helposti kohti videokuvan näyttämää suuntaa
- käyttämään robotin mahdollista "eliminointityökalua"
- hätäpysäyttämään robotti nopeasti
- sammuttamaan robotti

Käyttötilassa käyttäjän tulee lisäksi pystyä suorittamaan vastaavasti seuraavia *käyttöliittymän ohjaukseen liittyviä* toimintoja:

- merkitsemään karttaan (hiirellä) havaitsemansa kiinnostavat kohteet
- liittämään karttaan merkittyihin kohteisiin tekstitietoa
- kytkeämään joko videokuvan tai karttakuvan koko näyttöruudun kokoiseksi
- kytkemään päälle tai pois (näkyviin tai pois) erilaiset käyttöliittymän ilmaisimet tai toiminnot, kuten karttaan merkittyjen kohteiden tai niiden tekstitietojen näyttö
- zoomaamaan videokamerakuvaa (voi olla myös laitteisto-ohjaus, riippuu laitteistosta)
- vierittämään karttanäyttöä tunnetun alueen puitteissa, mikäli kartoitettava alue on laaja
- käynnistämään lokitoiminto, joka tallentaa robotin paikka- ja aikatiedot sekä mahdollisesti video- ja karttakuvat lokitiedostoon tai useisiin tiedostoihin

Mahdollisimman monille toiminnoille tehdään sekä graafinen, osoittimen avulla monitorilta tapahtuva että näppäimistöltä tapahtuva toteutustapa.

### 3 Käyttöliittymän toteutuksen testausjärjestelyt

Käyttöliittymää olisi hyvä päästä sen eri toteutusvaiheissa testaamaan niin käytännön toteutuskelpoisuuden kannalta (eli mitä käytössä olevalla laitteistolla on järkevää tehdä) kuin itse käyttöliittymän varsinaisen käytettävyyden kannaltakin. Arvatenkaan käyttökokeet robotin kanssa eivät alusta asti onnistu, joten käyttöliittymää laadittaessa on (turhan ohjelmointityön välttämiseksi) tehtävä riittävä määrä paperihahmotelmia. Ryhmän ulkopuolisten käyttäjien saaminen testejä varten siten, että testit olisivat mielekkäitä ei sekään ole välttämättä helposti toteutettavissa, joten suunnittelun aikana on keskityttävä täysillä hyödyntämään kaikki työryhmän sisältä löytyvä näkökulmadiversiteetti.

Käyttöliittymästä ja etenkin sen visuaalisesta layoutista on helppoa tehdä paperiprototyyppettä, hahmottelemalla kynän ja paperin kanssa. Sama pätee myös näkyvän käyttöliittymän pienempiin komponentteihin.

Toteutustehtävä on pakostakin luonteeltaan pienessä mitta- ja aikaskaalassa iteratiivinen, mutta olosuhteista johtuen ei mittaviin testaus- ja korjauskierroksiin ole mahdollisuutta. Heuristisissa testeissä turvaututtaneen ryhmän henkilöstöön. Osittain tämä on myös oikeutettua ottaen huomioon, että laitteen käyttäjäkuntakin on käyttökoulutuksen saanut väkeä. Testausjärjestelyitä hankaloittaa autenttisen stressitilanteen simuloimisen hankaluus.

### 4 Käyttöliittymän toteutuksen käytännön järjestelyt

#### Arvio suunnitelman toteutustasosta

Ensisijaisesti pyritään toteuttamaan robotin primäärit input- ja output-toiminnot eli robotin ja sen kamerapään ohjaus näppäimistöltä sekä robotin kuvaaman videokuvan reaaliaikainen tuottaminen tietokoneen ruudulle. Heti tämän jälkeen ovat toteutusprioriteetissa seuraavana robotin odometriadatan perusteella tehty paikannus ja kartoitus sekä näiden perusteella kartan sekä siinä liikkuvan robotin piirtäminen tietokoneen ruudulle. Kartoituksen toimiessa, pyritään siihen liittämään mahdollisuus kommentoida ja täydentää karttaa erilaisin merkinnöin. Edellä mainittujen toimintojen toteuttamista pidetään realistisena tavoitteena ja ainakin tähän pyritään. Näiden lisäksi on erinäisiä perusominaisuuksia, kuten hätäseis- tai sammutustoiminto melko helposti liitettävissä käyttöliittymään. Kamerapään ohjaus hiiren liikkeen avulla on myös yksi todennäköisesti toteutettava lisäominaisuus. Muut suunnitelmassa mainitut ominaisuudet ovat satunnaisessa järjestyksessä toteutettavien ominaisuuksien tärkeyslistalla.

#### Työnjako

Käyttöliittymän toteutus jaetaan seuraaviin osioihin, joiden toteutusvastuu jakautuu ohessa mainituille henkilöille:

- Projektipäällikkö (Käki)
- Käyttöliittymän suunnittelu (kaikki), paperiprotot ja -hahmotelmat (erityisesti Käki, Hacklin, Pohjoranta)
- Käyttöliittymän robotilta-käyttäjälle-suunnan visualisointi ja koodaaminen (Backman)
- Käyttöliittymän käyttäjältä-robotille-suunnan koodaaminen (Käki)

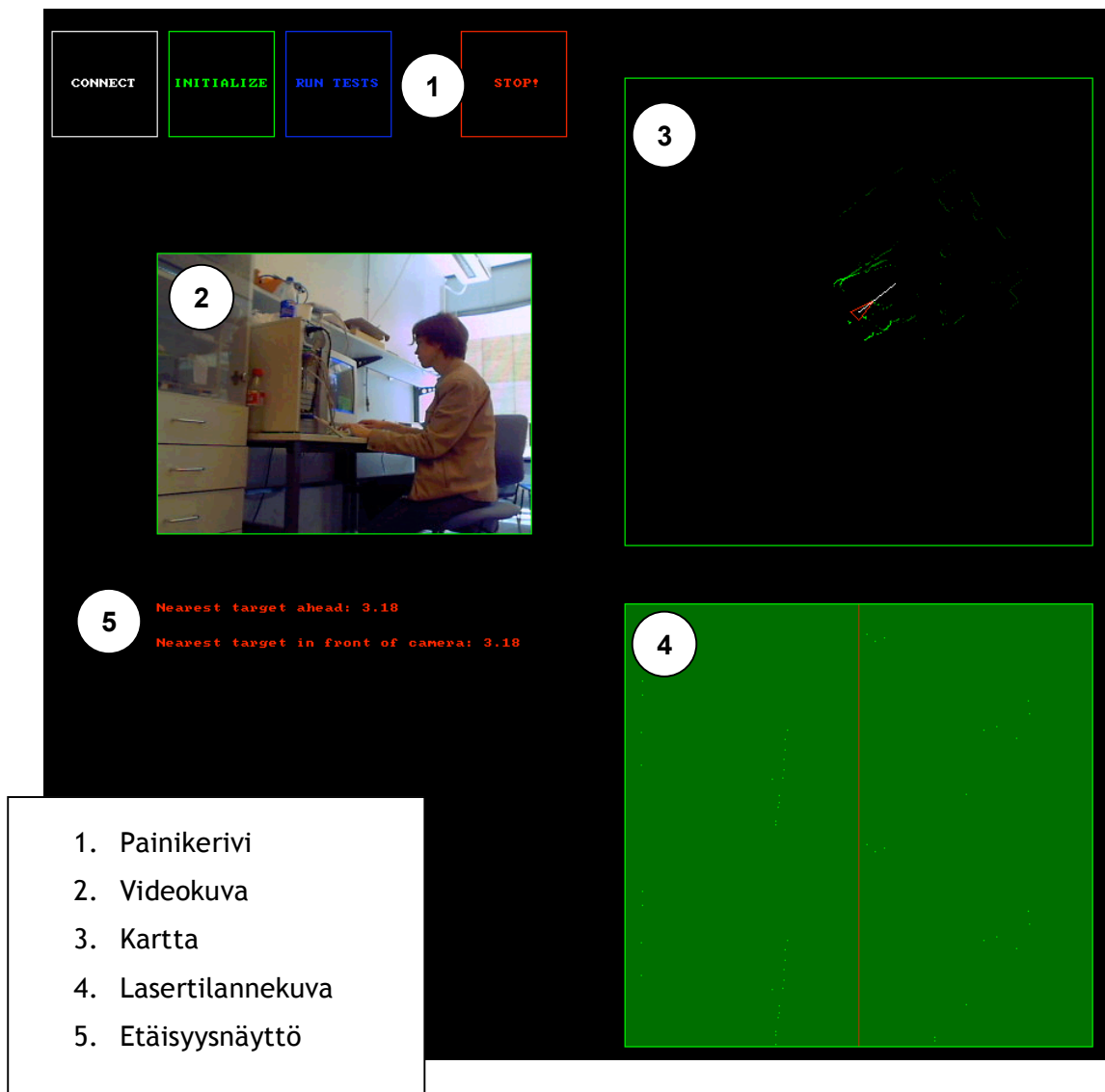
- Kartoitussjärjestelmän (SLAM) toteutuksen selvittäminen ja koodaaminen (Hackling, Pohjoranta)
- Odometriadatan mahdollinen suodatus ja analysointi (Pohjoranta)
- Konenäköelementtien mahdollinen toteutus (Pohjoranta)
- Käyttöliittymän testaus (Kaikki)

#### **Aikataulu (maalis–huhtikuu 2007)**

- 14.3. - Työn aloitus, ryhmäjako
- 26.3. - Robotin ja ohjelmointirajapinnan esittely
- 2.4. - 1. check-up, demonstointi ohjelmointirajapinnan osaamisesta
- 2.4. - Datanäytteitä datan käsittelyn ja kartoituksen suunnittelua varten
- 4.4. - Käyttöliittymäsuunnitelman esittely
- 4.4. - Ryhmätapaaminen
- 11.4. - Robotin ajomahdollisuus videokuvan perusteella
- 5.4.–12.4. - Pääsiäinen
- 13.4. - Kartoitustesti robotin kanssa
- NN.4. - Demonstraatio

# Ja mitä lopulta toteutui - visuaalinen käyttöliittymä

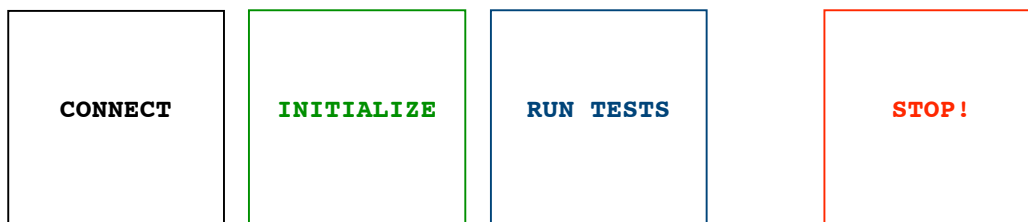
Varsinainen käyttöliittymä toteutti edellä hahmoteltuja vaatimuksia varsin hyvin - vaikka moni luonnosteltu päämäärä jäikin saavuttamatta, voi sanoa oikeastaan kaikkien *tärkeimpien* toimintojen tulleen toteutetuiksi. Lopputuloksena on yhden ihmisen käytettävä robotinkäskijä, joka pienellä koulutuksella ja totuttelulla sopii mainiosti labyrinttien tutkimisen komentokeskukseksi.



Kuva 1. Käyttöliittymänäkymä ja sen komponentit.

Robotin ajaminen tapahtuu oheisen näkymän (Kuva 1) avulla. Siinä olevat komponentit 2-5 on kaikki tarkoitettu sensoritiedon välittämiseen robotilta ajajalle. Vuorovaikutus kuljettajalta robotin suuntaan taas tapahtuu kokonaan näppäimistöllä ja hiirellä, joista lisää myöhemmin.

# 1 Painikerivi



Kuva 2: Toimintapainikkeet.

Ikkunan yläreunaa koristaa neljä toimintapainiketta. Ensimmäisestä, CONNECT-napista, muodostetaan yhteys J2B2-palvelimeen (yhteys ei siis nykyisellään synny vielä ohjelman käynnistyessä). Melko välittömästi edellisen jälkeen voidaan painaa INITIALIZE-nappia, joka alustaa varsinaisen robokontrollerin. Tämän jälkeen robotti onkin valmis ajoon.

RUN TESTS -painiketta käytettiin ohjelman ja robotin vuorovaikutuksen eri ulottuvuuksien kokeilemiseen silloin, kun varsinaista käyttöliittymää ei vielä ollut olemassa. Kilpailuversiona se on kuitenkin pelkkä koriste, samoin kuin viimeinen STOP!-nappi, joka lyö J2B2:n äkkijarrutuksen päälle. Saman lopputuloksen saa kuitenkin kätevämmän näppäimistön painikkeesta X.

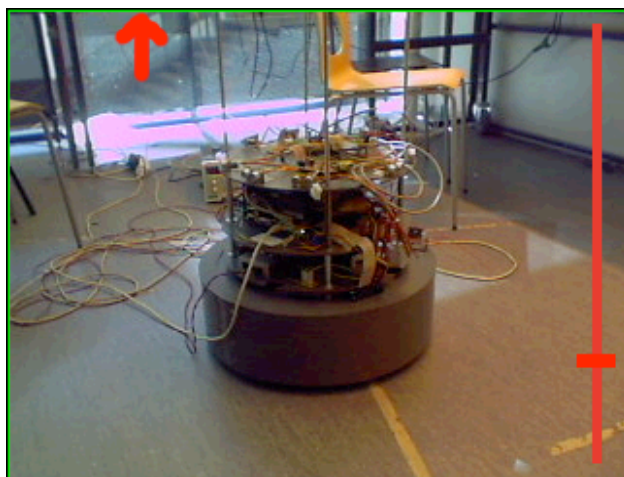
## Kehitettävää

Vaikka nappirivi näyttää visuaalisesti sängen vinkeältä toimintaruudun ylänurkassa, toiminnallisuudeltaan se on varsin yhdentekevä, ja niinpä sen voisi surutta poistaa kokonaankin näkymästä.

Itse asiassa napit tuovat nykyisellään käyttöliittymään yhden sen vakavimmista virheikäytön mahdollisuuksista: jos käyttäjä robottia käynnistäessään painaa käynnistyspainikkeita jotenkin muuten kuin ensin kerran CONNECT-nappia ja sen jälkeen kerran INITIALIZE-nappia, ohjelma saattaa kaatua ennen kuin se pääsee edes alkuun. Koska CONNECTin ja INITIALIZEn välissä ei koskaan ole tarpeen tehdä mitään muuta, olisi nämä kaksi toimintoa hyvä laittaa sekvenssinä yhden ja saman napin taakse - tai sitten suorittaa automaattisesti heti ohjelman käynnistyessä.

# 2 Videokuva

Lähes reaaliaikaisesti päivittyvä videoruutu näyttää maailman J2-B2:n silmin. Robotin tarjoama kuva on varsin pieni, mutta kuitenkin riittävän hyvä toimiakseen robotin ensisijaisena navigointinäkymänä. Käyttäessään kuvaa ajon apuna on syytä muistaa, että J2-B2:n ajosuunta saattaa poiketa jopa 90 astetta kameran suunnasta, eli katseltavissa on laaja 180 asteen sektori. Lisäksi kamera on käännettävissä pystysuunnassa miltei yhtä laajalla sektorilla, jotta



Kuva 3: Videokuvassa lajitoveri sekä pari karkeaa parannusluonnosta.



yläilmoissa ja varsinkin lattiatasossa vaanivat kohteet tulevat paikannetuiksi.

### Kehitettävää

Videokuvan parempi päivitysnopeus ja resoluutio parantaisivat luonnollisesti melko suoraan myös käyttökokemusta. Lisäksi kuvassa voisi näkyä vaikkapa pienellä suuntaansa vaihtavalla nuolella kuvan ylälaidassa se, miten kameran senhetkinen suunta suhtautuu robotin ajosuuntaan (ks. Kuva 3). Tämä helpottaisi pyöriteltävän kameran kanssa ajamista, vaikka periaatteessa (kohdassa 4 esiteltävä) lasertilanekuvakin ajaa pitkälti samaa asiaa. Ajamista kuitenkin helpottaa, jos kaikki olennaisin tieto on havaittavissa yhdellä silmäyksellä. Myös jonkinlainen ilmainen kameran pystysuuntaisesta orientaatiosta olisi avuksi. Tällaisena voisi toimia vaikkapa yksinkertainen pystyviiva kuvan jommassakummassa laidassa. Siinä liikkuisi pieni sivupalkki (ks. jälleen Kuva 3), joka keskikohdan ala- tai yläpuolella ollessaan ilmaisisi, että kamera on suunnattu muualle kuin suoraan vaakatasoon.

## 3 Kartta

Kartta muodostetaan robotin laserodometrian perusteella, ja se merkitsee kaikki laserin paikantamat esteet yksinkertaisesti vihreillä pisteillä. Vihreä väri voimistuu tiettyyn pisteeseen asti sen mukaan, mitä useampia mittauksia kyseisestä esteestä on saatu.

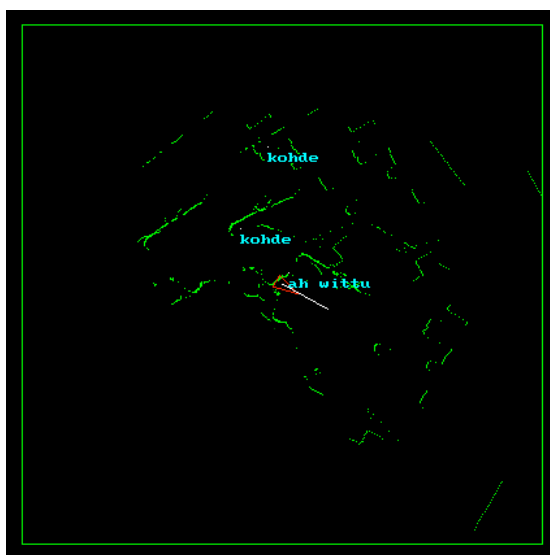
Karttanäkymän keskellä on aina itse robotin sijainnin ilmaiseva punainen kolmio (jonka terävin kärki näyttää robotin ajosuuntaan) sekä valkoinen suuntavektori, joka näyttää, mihin kamera on suunnattuna.

Karttaan merkitään myös robotin paikantamat, käytännössä ajajan videokuvasta tunnistamat "vaaralliset" tai muuten kiinnostavat kohteet. Suunnitteluvaiheessa pohdittiin myös tämän tunnistuksen automatisoimista (esimerkiksi liikkuvien kohteiden tapauksessa) mutta lopulta toteutui ainoastaan manuaalinen kohteenosoitus hiirellä. Kohteet merkitään pienellä sinisellä pisteellä ja tekstillä "kohde". Lisäksi kerran tehtävän aikana on mahdollista merkitä robotin senhetkinen sijainti talteen pikanäppäinkomennolla H, jolloin siihen kohtaan kartalle ilmestyy teksti "ah wittu". Tätä voi käyttää esimerkiksi kesken kiivaan taistelun paikannetun terroristijohtajan komentopaikan merkitsemiseen.

### Kehitettävää

Kohteiden merkitsemiseen tarjottu nimiavaruus ei vielä päätä huimaa. Niinpä varsinkin jos kohteita on monia erilaisia, olisi hyvä lisätä käyttöliittymään mahdollisuus kirjoittaa itse kohdetta kuvaava teksti. Tämän ohella tai jopa asemesta erilaisille kohteille voisi olla tarjolla muutama erilainen väri vaihtoehto.

Projektin alkuvaiheessa mietimme sitäkin mahdollisuutta, että robottia voisi ohjata tavanomaisen näppäimistöajon lisäksi klikkaamalla hiirellä jo kartoitetulta alueelta paikkaa, johon robotti sen jälkeen huristelisi, joko suoraan (jolloin käytännössä



Kuva 4: Kartta. Robotti, esteitä ja muutama merkitty kohde.

kohteen pitäisi olla esteettömän linjan päässä nykysijainnista) tai mahdolliset esteet reitinhakutekoälyllä kiertäen. Ainakin ensimmäinen vaihtoehto olisi varsin suoraviivainen (kirjaimellisestikin) toteuttaa, ja jo tutkittujen alueiden läpikulusta saisi sillä hieman nykyistä vaivattomampaa.

Pohdimme myös vielä tutkimattoman alueen esittämistä eri värillä kuin robotin jo havainnoiman maaston. Lähinnä tästä olisi hyötyä havaintoalueen reunoilla, joilla (esimerkiksi Kuva 4 oikealla alhaalla, kuten alakulmaan paikannetusta esteestä huomaa) voi olla laserilla havaittuja avoimia tiloja, joita ei helposti hahmota sellaisiksi. Vaikea sanoa, miten hyvin tämä toimisi yhteen muun odometriadatan visualisoinnin kanssa - olisiko lopputulos nykyistä selkeämpi vai sekavampi.

Itse kilpailutilanteessa, varsinkaan ajon loppupuolella, kartta ei enää pysynyt synkassa robotin todellisen sijainnin ja ympäristön kanssa. Syy tähän oli yksinkertainen: joku (todennäköisesti hyökkäyksemme yllättämän terroristisolun edustaja) siirsi robottiamme useampaan otteeseen sen juututtua puskemaan jotakin labyrinthialtaan seinää päin. Siirtely oli epäilemättä robon omaksi parhaaksi, mutta odometria ei pysynyt yllättävien origonmuutosten perässä, ja jo paikannetut kohteet karkasivat kartalla hämmentäviin paikkoihin. Ehkä lasermittauksista voisi olla tunnistettavissa myös tällainen tilanne, jossa robotti de facto siirtyy vaikka sen pyörät eivät rullaakaan minnekään (tai ainakaan siirtymissuuntaan), mutta mitenkään helppoa sen ohjelmointi ei liene.

## 4 Lasertilannekuva

Tilannekuva esittää periaatteessa samaa odometriadataa kuin karttakin, mutta paljon pienemmässä mittakaavassa, joten sen mukaan ajaminen on selvästi käytännöllisempää. Kuvan ylälaita on aina kohti ajosuuntaa. Kameran suunta ilmaistaan kuvan alareunan keskeltä lähtevällä punaisella suuntavektorilla, joka voi siis äärimmillään osoittaa 90 astetta jommallekummalle sivulle. Käytännössä ajamisesta tuleekin helposti videokuvan ja odometrianäytön vilkuilua vuoronperään.

### Kehitettävää

Kuten kilpailuajossakin tuli karvaasti todennettua, robotti ei ole pistemäinen vaan kappale, jolla on ulottuvuuksia. Tästä seuraa muun muassa se, että robotti ei mahdu etenemään mielivaltaisen ahtaista väleistä, vaan tarvitsee itselleen sen noin 40 cm:n tilan. Tätä voisi havainnollistaa myös lasertilannekuvassa piirtämällä robotinkokoisen ympyrän (tai osan sitä) kuvan alalaitaan, jotta esteiden ja vapaan ajotilan arviointi helpottuisi. Esteiden näytön ei myöskään tarvitsisi rajoittua robotin etureunassa olevan laserskannerin tasolle, vaan odometrianäyttö voisi muistaa maastoa hieman tämän tasan taaksekkin. Tällöin kääntyminen ja uuteen suuntaan ajaminen helpottuisi.

Varsin merkittävä, mutta teknisesti vaikea ongelma on, että laserskanneri tutkii ympäröivää maastoa noin puolen metrin korkeudelta ja jättää näin ollen havaitsematta lattiatasossa piilevät matalat, mutta robotin etenemisen kannalta yhtä kaikki fataalit esteet. Jo tämänkin vuoksi pelkän laserin perusteella ajaminen ei kannata, vaan kameran aktiivinen käyttäminen eri suuntiin myös pystytasossa on hyvin suositeltavaa.

Kuten suuremman mittakaavan kartasta, myös tästäkin näkymästä puuttui eksplisiittinen mittakaavanilmaisus, jollaisesta olisi hieman apua etenkin videokuvan tarkempaan tarkasteluun. Sellaisen lisääminen ei olisi iso homma.

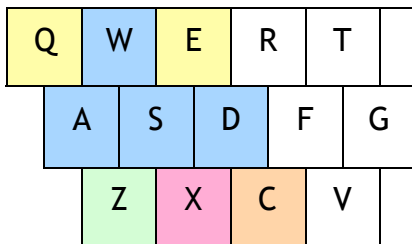
## 5 Etäisyysnäyttö

Videokuvan alla olevat kaksi pientä riviä ovat itse asiassa erittäin tärkeitä: ne kertovat laserodometrian ilmoittamat etäisyydet (metreinä) lähimpään kohteeseen robotin etenemis- ja kameran kuvaussuunnassa. Ne tuovat kamera-ajoon lisää syvyyttä (kirjaimellisestikin), sillä etäisyyksien silmämääräinen arviointi antaa varsinkin vieraita huoneistoja tutkiessa usein vääriä käsityksiä tilan ulottuvuuksista. Ennen kaikkea etäisyysmittari saattaa auttaa huomaamaan seinän tai muun esteen ennen kuin robotti rymistelee sellaista päin.

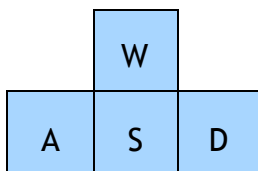
### Kehitettävää

Nykyisellään luvut eivät viestitä tehokkaimmalla mahdollisella tavalla varsinkaan pieniä etäisyyksiä (eli törmäyksen uhkia). Värikoodaus auttaisi tässä paljon: esimerkiksi vihreä väri yli metrin etäisyyksille, keltainen matkoille puolen metrin ja metrin väliltä, ja punainen vaaran väri alle puolessa metrissä vaaniville uusille tuttavuuksille. Ehkä luvut olisivat myös tehokkaammin luettavissa videokuvan (tai odometriaruudun) osana kuin tuossa erillisenä pienenä kokonaisuutena.

## 6 Näppäimistö



Robotin ajaminen tapahtuu kokonaisuudessaan näppäimistöltä. Käytössä ovat seuraavat toimintapainikkeet:



Perinteinen 3D-tietokonepeleistä tuttu nuoliohjain. **W** ajaa robottia eteenpäin, **S** peruuttaa. **A** pyörittää robottia vastapäivään (vasemmalle), **D** myötäpäivään (oikealle).



Pikakurkistus. **Q** kääntää robotin kameran 90 astetta vasempaan ja pitää sen käännettynä niin kauan kuin nappi on alhaalla. Napin vapautuessa kamera palaa entiselleen. **E** toimii samoin, mutta päinvastaiseen suuntaan.



**Z** "resetoi" kameran eli palauttaa sen katsomaan suoraan eteenpäin robotin ajosuuntaan.



**X** saa robotin tekemään jarrituksen. Hyödyllinen erityisesti yhdistettynä tarkkaavaiseen etäisyysnäytön seurantaan.



**C** vaihtaa hiiriohjauksen kahden eri moodin välillä. Ensimmäisessä moodissa hiiri ohjaa kameraa, toisessa moodissa taas kamera pysyy paikallaan ja hiirtä voi käyttää kohteiden merkitsemiseen kartalle.

Näppäimistöohjaus toimii kaiken kaikkiaan hyvin notkeasti ja intuitiivisesti. WASD-nuoliohjaus on tietokonepelejä pelanneilla lähes selkäytimessä, ja pikakurkistusnapit “luonnollisilla” paikoillaan nuolinäppäinten vieressä. Myös kolme muuta painiketta sijaitsevat ohjaimen välittömässä yhteydessä, ja vilkas mielikuvitus löytää helposti niihin myös pienet muistisäännöt: C niin kuin camera kytkee kääntyvän kameran päälle ja pois, X luonnollisesti pysäyttää toiminnan ja Z kumoo kameralle tehdyt kääntelyt palauttaen status quon (kuin Ctrl+Z).

### **Kehitettävää**

Vaikka näppäinohjaus on tuollaisenaan hyvin toimiva, robotti lähtee liikkeelle varsin suurella viiveellä siitä, kun W-näppäintä on painettu. Tälle ei kuitenkaan liene käyttöliittymäpäässä ihmeitä tehtävissä, vaan hitaus löytynee rajapinnan toiselta puolelta, palvelimen ja robotin päästä. Ehkä.

## **7 Hiiri**

Hiirtä käytetään edellä mainitun kohteiden osoituksen ja merkinnän (sekä ylärivin koristenappien koskettelun) lisäksi, ja ennen kaikkea, kameran kääntelyyn. Hiiren kuljetus vasemmalle kääntää kameraa vaakatasossa vastapäivään eli vasempaan, kun taas siimahännän raahaus oikealle kiepauttaa kameraa pystyakselinsa ympäri myötäpäivään kohti oikeaa. Pystyohjauskin toimii intuitiivisesti, ainakin jos on joskus harrastanut lentämistä tai edes lentosimulaattoreita - käyttäjä kallistaa siis kameraa yläviistoon vetämällä hiirtä itseään kohti ja alaviistoon työntämällä hiirulaisen itsestään loitommas.

### **Kehitettävää**

Hiiri svengaa kuin hirvi, ei sinänsä valittamista. Mutta jos kohteiden merkitsemistä kartalle haluaa hieman kehittää, on siinä todennäköisesti myös hiirelle tiedossa uusia tehtäviä esimerkiksi kohteiden tyyppin (eli merkintävärin) valinnassa.

## **8 Muuta mainittavaa**

Roboliittymän tekninen toimivuus on hieman kaksijakoista: ohjelma lähtee käyntiin vikkelaasti, mutta ajon aikana se rohmuaa melkoisesti (ja koko ajan kasvavan määrän) muistia, ja ohjelman sulkeminen löi tyypillisesti alustana käytetyn Windows XP:n ja DevC++:n kyykkyy jopa minuuteiksi. Kovin pitkiä löytöretkiä robotilla ei siis nykyisen käyttöliittymän komennossa kannata tehdä.

Piirretty kartta ei nykyisellään myöskään tallennu mihinkään - jokin tapa jo tehdyn kartoitustyön ja paikannettujen kohteiden ikuistamiseksi olisi selkeä edistysaskel. Myös suunnitelmassa mainittu videon tai edes still-kuvien, erityisesti kohteista nappattujen otosten, tallennus kovalevylle olisi antiterrorirobotin todellisessa käyttötilanteessa arvokas lisäfeature.

Niin, ja se “eliminointityökalu” olisi tietysti kerrassaan tarpeellinen höyste robotin repertuaariin, kun mennään oikeasti vaarallisiin paikkoihin.

# Reflektiota ja kurssipalautetta

## Harjoitustyöstä yleisesti

Harjoitustyö on hyvä - sen fokusta voisi kuitenkin siirtää enemmän itse käyttöliittymän suunnittelun ja toteutuksen harjoitteluun puhtaasta koodaamisen ja SDL-yms. UI-kirjastojen opettelusta. Tämä voisi onnistua antamalla valmiita (koodi)esimerkkejä siitä, miten erilaiset käyttöliittymäkomponentit toteutetaan.

Lisäksi harjoitustyöhön olisi hyvä ottaa mukaan myös suunnitteluun kuuluvat pahvimalli- ja käyttäjättestausvaiheet (myös näiden raportointi).

Kurssin esitietovaatimuksissa tulisi mainita, että kurssin suorittaminen edellyttää faktisesti C++ -ohjelmakielen hallintaa ja rutiinia. Ryhmämme hyvin eritasoiset C++ -valmiudet johtivat toteutuksen työmäärän hyvin epätasaiseen jakautumiseen. Osaavammat tekivät kiitettävän urakan koodin äärellä, kun taas "projektipäällikön" homma painottui alustavan konseptoinnin ohella lähinnä lopputuloksen hämmästelyyn ja dokumentointiin sekä ajokortittoman kuskin vakuuttavuudella hoidettuun kilpailuajoon, josta roboparka onneksi taisi selvitä päällisin puolin ehjänä.

## Luennoista

Teorialuennot olivat nyt melko väljiä - niiden ohella voisi teettää opiskelijoilla vaikka esseen käyttöliittymän suunnittelun teoreettisista asioista, jotta näihin tulisi tutustuttua riittävän syvällisesti. Toinen vaihtoehto olisi keskusteluluento, jossa esim. ruodittaisiin jotain käyttöliittymäkeiksiä peilaten teorian käsitteisiin ja metodiikkaan - siis konkreettisemmalla tasolla kuin, että "ipod on hyvä, siemens gigaset ei ole", vaikka opettavaista se hyvien ja pahojen käyttöliittymäpiirteiden analysointi oli. Vierailuluento (Tamrockin porausmasiina) oli antoisa.

## Dokumentaatiosta

Kuten assistentit totesivat, mikään määrä dokumentaatiota harjoitustyössä käytettävän laitteiston ja koodin suhteen ei tunnu ikinä olevan tarpeeksi. Olisi siis syytä ehkä miettiä, miten laitteiston dokumentaatio ja esittely toteutetaan siten, ettei aukkoja ole systeemin perustiedoissa kun harkkatyötä lähdetään tekemään - dokumentaatiosta pitäisi tehdä tarkoituksenmukaisempaa ja paremmin ko. harjoitukseen räätälöityä. Robotin dokumentissa tulisi lukea tarkalleen ne tiedot, joita tarvitaan karttasovelluksen laatimiseen jne... Ja koodissa tarkalleen ne tiedot, joita tarvitaan UI-komponenteissa... Kaikki ylimääräinen vain hämää ja sekoittaa.

## Ohjelmointirajapinnan toteutuksesta

Päällimmäisinä mieleen jäivät robon nopeuden ja kääntymisen komentamisen hankaluudet; robo menee jumiin, jos sille komennettujen kulma- ja lineaarinopeuksien summa ylittää jonkin raja-arvon. Lisäksi kääntyminen tosiaan annetaan kulmanopeuden muutoksena, eikä esim. määränä käänösasteita tms. Näitä vaihtoehtoja voisi olla ihan hyvin olemassa valmiina ohjelmointirajapinnassa.

## Työmäärästä

Kuten edellä todettiin, työmäärä ei jakautunut ryhmän sisällä aivan tasaisesti, mutta yksittäisten jäsenten työpanokset olivat arviolta 25-40 tunnin luokkaa.