

Robottiohjelmoinnin harjoitustyö

VahtiRobo

Kristiina Rahikainen
kristiina.rahikainen@cs.helsinki.fi

VahtiRobon kuvaus

VahtiRobo on robotti, joka liikkuu sille asetetulla reviirillä ja hyökkää kaiken reviirillä olevan kimppuun. Robotti erottaa reviirin rajan maahan osoittavalla valosensorilla, joten rajan pitää olla merkitty niin, että sillä on kontrastia reviirin sisäosaan nähden. Ennen kuin VahtiRobo alkaa vartioda reviiriä, se pyytää mittaamaan reviirin pohjaväriä ja reunojen väriä. Robotti liikkuu eteenpäin reviirillä ja törmätessään rajaan se peruuttaa ja kääntyy n. 120 astetta oikealle. Havaitessaan jotain reviirillä, se syöksyy sitä kohden ja törmätessään siihen lyö saksillaan. Kovan äänen kuullessaan robotti kääntyy katsomaan ympärilleen, mikäli se on äänen kuullessa ollut liikkumassa normaalisti eteenpäin.

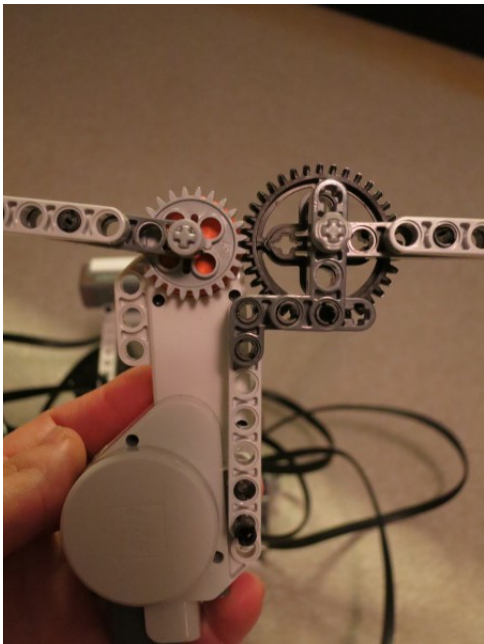
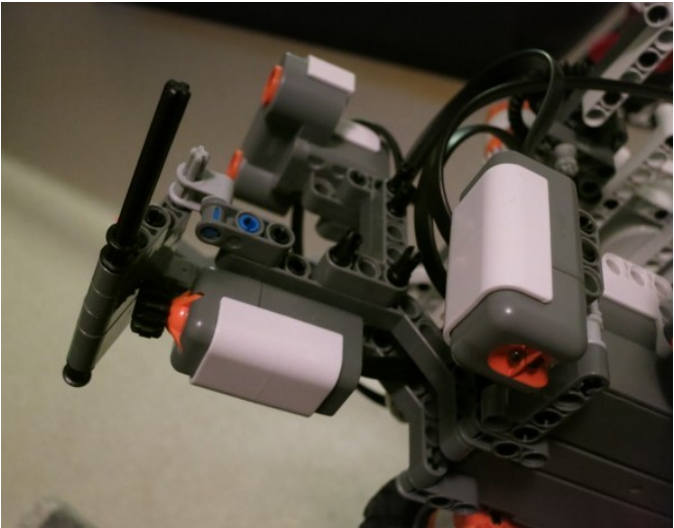
Robotin rakenne

VahtiRobo kulkee kolmella pyörällä, joista kaksi on kiinni moottoreissa. Ultraäänisensori osoittaa eteenpäin ja sen edessä on myös kosketussensori ja maahan osoittava valosensori. Robotin keskiosan päällä on tukirakenne, joka kannattelee äänisensoria sekä moottoria, joka liikuttaa robotin saksia.

Mitat:

- Renkaiden väli: 16 cm
- Renkaiden halkaisija: 5,6 cm





Koodin rakenne

Koodi koostuu yhdeksästä luokasta.

VahtiRobo käynnistää ohjelman ja Guardianin valmistelun.

UI pyytää käyttäjää antamaan valosensorin arvot reviirin sisältä ja sen rajalta.

ValueSetter lukee valosensorin antamat arvot ja laskee niiden perusteella Guardianille reviirin rajan valoarvon liikkumatilalla.

Guardian luo käyttöluokat ja antaa ne Arbitrator-luokan ilmentymälle ajettavaksi.

Käyttöluokat toteuttavat rajapinnan Behavior, ja niitä on viisi: Drive, HeardSomething, Charge, Turn ja Attack.

AbortMission hoitaa robotin hätäpysäytyksen.

Ohjelma käynnistyy VahtiRobon mainista. Aluksi luodaan AbortMission omaan säikeeseensä, jotta hätäpysäytys toimisi kaikissa tilanteissa. Sitten luodaan Guardian metodilla setUpGuardian(), joka luo robotin liikkumiseen käytettäviä moottoreita hallitsevan DifferentialPilotin ja sensoreita käyttävien luokkien ilmentymät sekä Guardianin. Seuraavaksi setUpGuardian() luo UI:n ja ajaa sen metodin askValues(), joka pyytää käyttäjää ensin painamaan vasenta nuolinäppäintä reviirin sisällä ja sitten oikeaa reviirin rajan päällä.

UI kutsuu tällöin ValueSetterin metodeja setBackgroundLightValue() ja setBorderLightValue(), jotka asettavat arvot ValueSetterille. Sen jälkeen UI kutsuu ValueSetterin metodia setValuesForGuardian(guardian) parametrina sen luoma Guardian. ValueSetter laskee sopivan rajan valoarvon metodilla getBorderValue() ja asettaa Guardianille sekä rajan valoarvon, että booleanin joka kertoo onko reviirin raja tummempi kuin reviirin sisäosa. UI palauttaa VahtiRobolle Guardianin ja VahtiRobo ajaa sen.

Guardian luo käyttöluokat ja antaa ne Arbitratorille tärkeysjärjestyksessä. Alimpana on Drive, joka ajaa robottia eteenpäin. Seuraavana on HeardSomething, joka ottaa kontrollin, jos robotin äänisensori kuulee kovan äänen. Silloin robotti tarkistaa kääntymällä ensin oikean puolensa ja sitten vasemman. Jos robotin ultraäänisensori havaitsee jotain, ottaa käytös Charge kontrollin. Tällöin robo soittaa tiedoston paras2.wav, nopeuttaa vauhtiaan ja lähtee eteenpäin. Jos robotin valosensori osuu reviirin rajalle, käytös Turn ottaa kontrollin ja robotti ensin peruuttaa ja sitten kääntyy n. 120 astetta. Tärkeimpänä käytöksenä on Attack, joka ottaa kontrollin mikäli robotin kosketussensoria painetaan. Attack pysäyttää robotin ja hallitsee robotin kolmatta moottoria, jossa on kiinni robotin "sakset". Moottoria käännetään ensin n. 45 astetta myötäpäivään ja sitten saman verran vastapäivään. Sen jälkeen robotti peruuttaa n. 5cm.

Robotti jatkaa käytöstään kunnes käyttäjä painaa enter-nappia, jolloin AbortMission lopettaa ohjelman ajamisen.

Testaus

Olenaisia testitapauksia on ottaa selville tunnistaako robotti reviirin sisäosan ja rajan erilaisissa ympäristöissä, ja että robotti pysyy reviirin sisällä riippumatta reviirin muodosta ja siitä mikä käytös robotilla on aktiivisena rajalle tullessaan. Olennaista on myös selvittää minkälaiset tunkeutujat

robotti reviiirillään huomaa.

Testitapaus 1:

Testatin robottia tummalla lattialla reviiirin rajoina valkoinen paperi. Robotti kysyi ensin reviiirin sisäosan sekä rajan valoarvoa ja annettuani ne robo alkoi ajaa reviiirin sisällä ja kääntyi aina raja-
osuessaan.

Testi onnistui: robotti kääntyi aina rajalle tullessaan.

Testitapaus 2:

Testatin robottia vaalealla lattialla reviiirin rajoina musta sähköteippi. Robotti kysyi ensin reviiirin sisäosan sekä rajan valoarvoa ja annettuani ne robo alkoi ajaa reviiirin sisällä ja kääntyi aina raja-
osuessaan.

Testi onnistui: robotti kääntyi aina rajalle tullessaan.

Testitapaus 3:

Testattu robotin hyökkäystä. Asetettiin robo liikkumaan alueelleen ja laitettiin vähän matkan päähän sen eteen. Robotti syöksyi raivokkaasti eteenpäin ja törmätessään käteen raapaisi saksillaan sekä peruutti sitten. Testi onnistui.

Testitapaus 4:

Laitettu robotti liikkumaan ja vihellettiin kovaa sen vieressä. Robo lähti kääntymään ympäri ja esteen nähdessään lähti syöksymään kohti. Testi onnistui.

Testitapaus 5:

Laitettu robotti liikkumaan alueelle ja laitettu käsi robon näköpiiriin, mutta reviiirin ulkopuolelle. Robo lähti syöksyyn, mutta tullessaan rajalle pysähtyi ja käästyi ympäri. Testi onnistui.

Testitapaus 6:

Laitettu robotti liikkumaan itseksensä alueelle. Raja-
törmättyään robotti kääntyi ympäri ja jatkoi matkaansa. Liian loivassa kulmassa oikea kylki raja-
osuessaan robo kuitenkin kääntyessään joutui reviiirinsä ulkopuolelle ja jatkoi siellä matkaansa, testi onnistu siis vain osittain.

Ongelma johtuu siitä, että valosensori sijaitsee robon vasemmalla puolella ja käännöksen aikana se meni rajan väärälle puolelle. Robotin rakenteen muuttaminen siten, että valosensori on aivan keskellä voisi auttaa ongelmaan. Toinen valosensori robotin oikealla puolella voisi myös auttaa, jos oikean sensorin osuessa raja-
osuessaan robotti kääntyisi vasemmalle, ja vasemman sensorin osuessa raja-
osuessaan oikealle.

Rajoitukset ja tulevaisuus

Robotti toimii vain rajatulla, tyhjällä reviiirillä. Jos reviiirillä on robotin lisäksi jotain muita esineitä, robotti hyökkää niiden kimppuun. Ultraäänisensori ei huomaa kaikkia kohteita, jos ne ovat liian matalia tai kapeita ja pehmeitä. Kosketussensori ei myöskään huomaa kohteita, jotka ovat alas-
paljon kapenevia, kun robotin yläosa törmää kohteeseen, mutta kosketussensori ei.

Tulevaisuudessa robotti voisi myös esim. kartoittaa reviiirinsä ennen partioinnin aloittamista, jotta se

tietäisi mitkä esineet ja esim. seinät voivat olla sen reviirillä ilman, että niiden kimppuun tarvitsee hyökätä.

Käyttöohje

Kun olet koonnut robotin, varmista mittasuhteiden paikkansapitävyys ja aja build.xml kun robotti on kiinnitetty koneeseen. Robotti tarvitsee reviirikseen alueen, jonka väri on suht tasainen ja joka on rajattu toisella värillä, jolla on kunnon kontrasti reviirin väriin nähden. Esim. vaalea lattia rajattuna mustalla sähköteipillä toimii.

1. Ohjelma käynnistyy VahtiRobo-tiedostosta. Käynnistä ohjelma ja aseta robotti reviirinsä sisälle, niin ettei valosensori ole rajan päällä. Paina robotista vasenta nuolinäppäintä näytön ohjeiden mukaan.
2. Siirrä robotti niin, että valosensori on suoraan reviirin rajan päällä ja paina robotista oikeaa nuolinäppäintä, kuten näytöllä ohjeistetaan.
3. Robotti kääntyy nyt pois rajalta ja lähtee sitten liikkumaan reviirillä! Se partioi aluetta, kunnes enter-nappia painetaan.