

# **Robottiohjelmoinnin Harjoitustyö**

**Laby**

Nico Roos  
014428092  
[Nico.Roos@cs.helsinki.fi](mailto:Nico.Roos@cs.helsinki.fi)

## **Labyn Kuvaus**

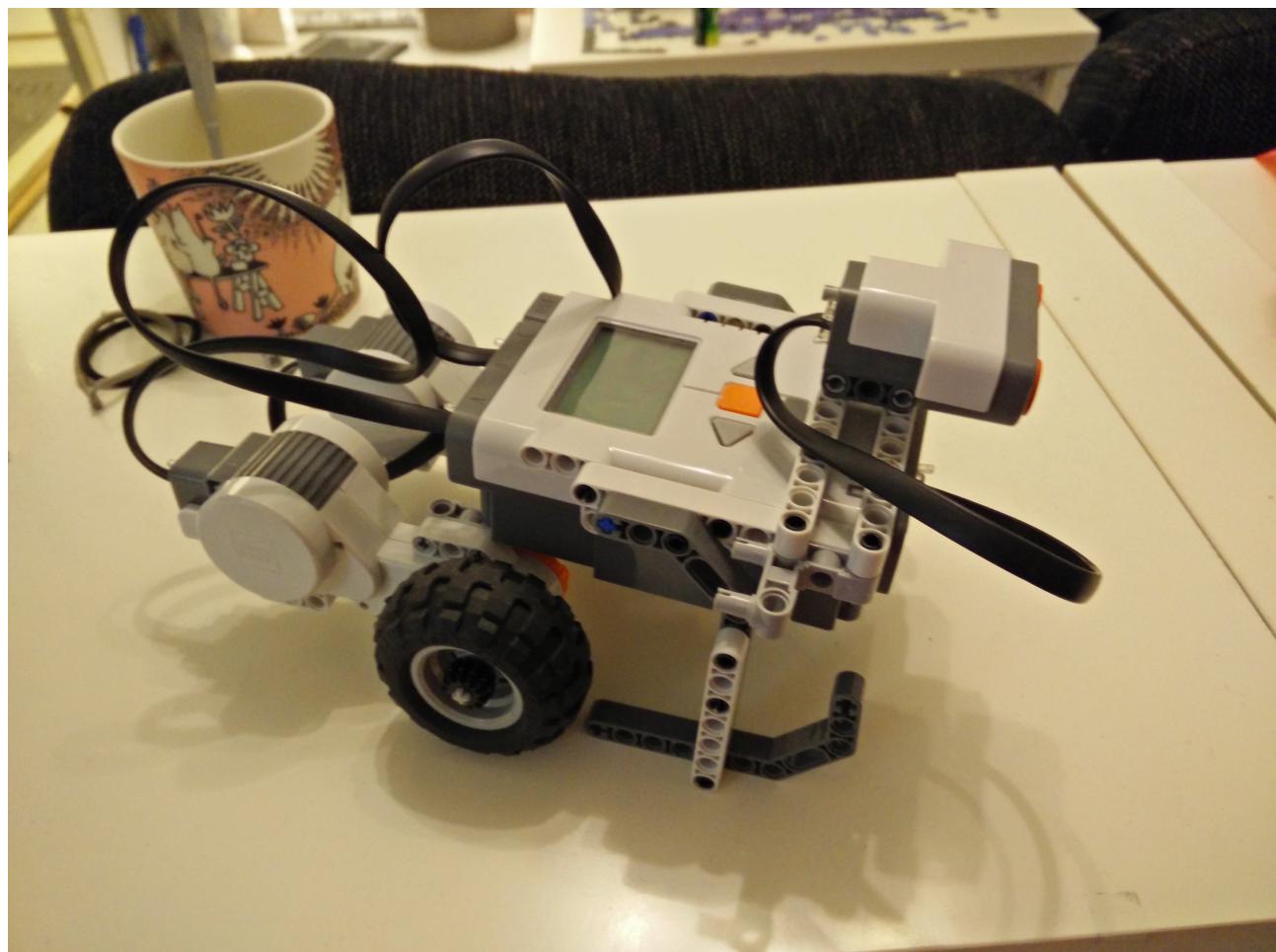
Labyn tehtävä on suoriutua ulos labyrintista. Labyrintin tulee koostua "ruuduista" jotka ovat n. 20Cmx20cm kokoiset. Jokaisella ruudulla voi olla 0-4 seinää joiden tulee olla 90° asteen kulmassa toisiinsa nähdien. Robotti simuloi syvyshakua ja tarkistaa jokaisessa ruudussa mahdollisen reitin järjestyksessä vasen > eteenpäin > oikealle ja valitsee aina ensimmäisen mahdollisen. Ajautuessaan umpikujaan robotti peruuttaa samaa reittiä jota tulikin, tarkistaessaan joka ruudun kohdalla onko vielä käymättömiä reittejä (Jos viimeksi käännyttiin vasemmalle, on vielä mahdollisuus että päästään suoraan tai oikealle jne.).

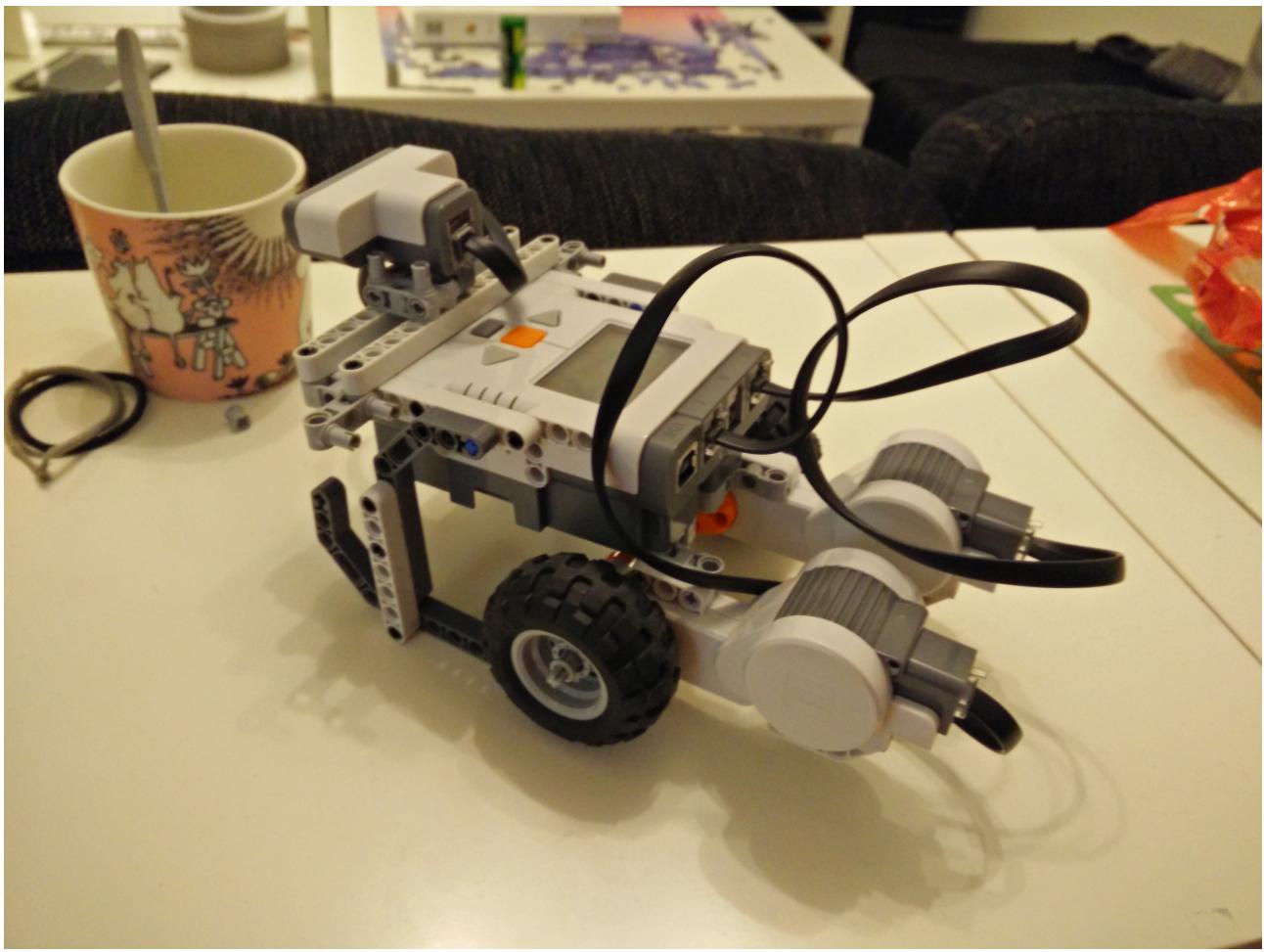
Labyrinttia rakentaessa tulee olla huolellinen seinien välimatkoista ja suorista kulmista robotin toiminnan varmistamiseksi.

Robotti on siitä "tyhmä" että se ei tiedä itse löytäneensä ulos labyrintista. Yritin keksiä tapoja robon ymmärtää itse löytäneensä ulos, mutta kaikki mahdollisuudet rajoittivat itse labyrintissa toimimista liikaa (esim. Jos joka suunnassa on tyhjää == ollaan maalissa, rajoitaisi käytävän pituuksia labyrintissa)

## **Robotin rakenne**

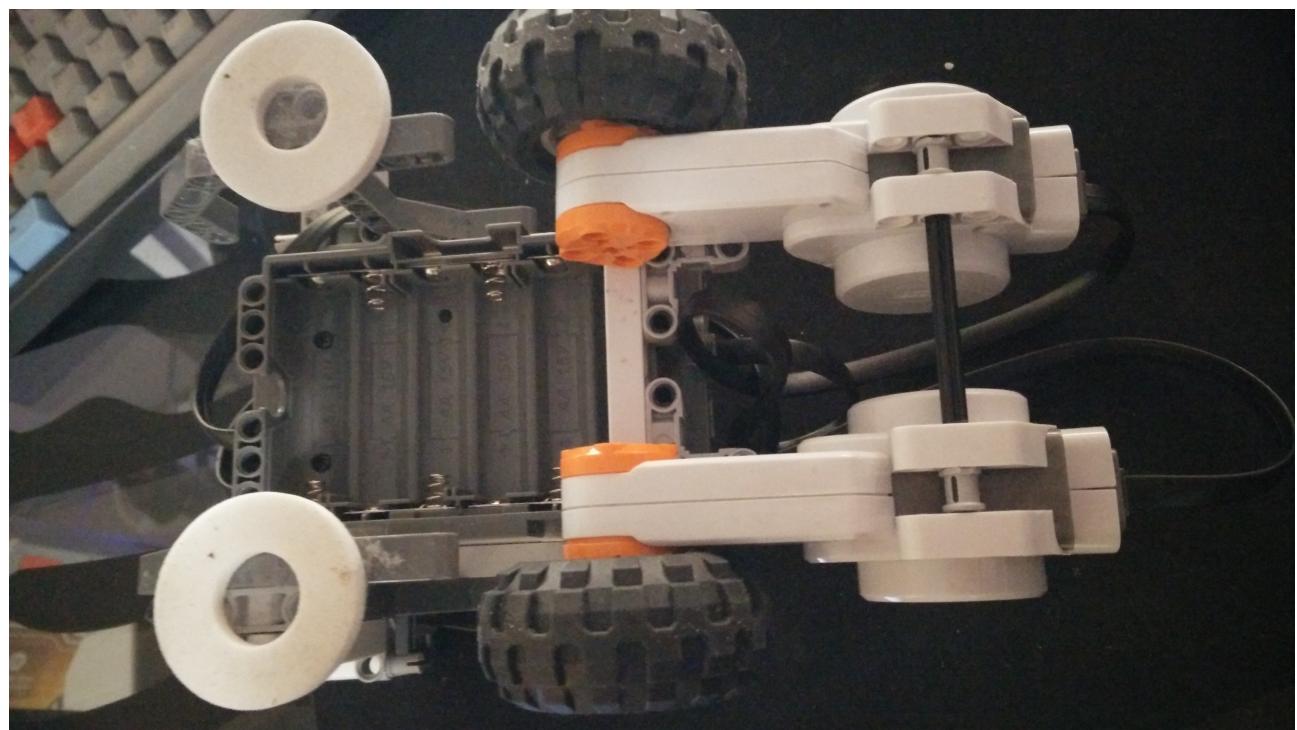
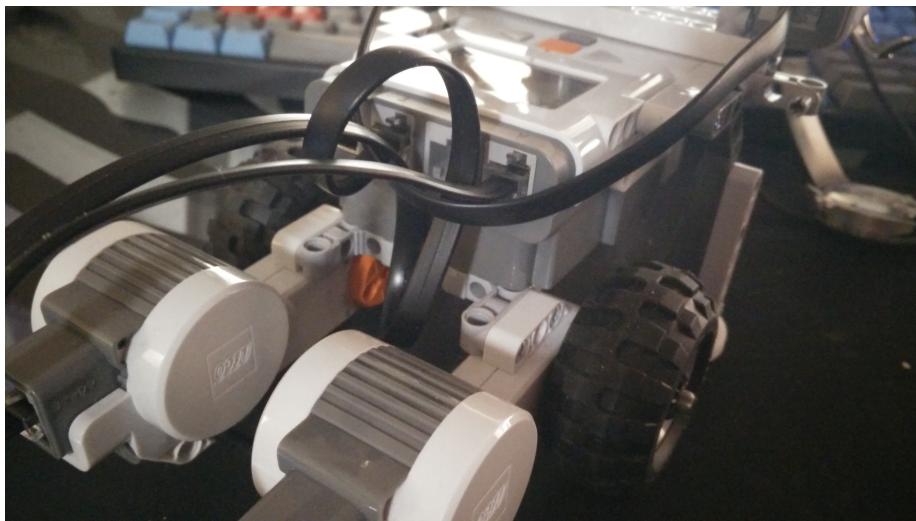
Laby on noin 20senttiä pitkä ja sen akseliväli on 11.6cm ja renkaan koko 5.6. Nämä arvot tulee laittaa DifferentialPilotille tiedoiksi. Robotilla on kaksi moottoriohjattua rengasta sekä kaksi tassuilal varustettua jalasta pitämässä sitä pystyssä. Sensorille olennainen paikka on robotin etuosassa, sen päällä.





Kahdessa ensimmäisessä kuvassa tassut eivät ole vielä kiinni





## Koodin rakenne

Labyn koodi koostuu kahdesta luokasta. Pääluokkana toimii Laby.java joka pyörittää mainLoop():ia ja luo sensorit sekä päätää mennäänkö eteenpäin, taaksepäin vai lopetetaanko kokonaan.

MainLoop:in sisällä on while silmukka joka pyörii niin kauan kunnes kosketussensoria pidetään pohjassa silmukan alussa. Silmukasta päästään myös ulos mikäli huomataan että labyrintistä ei ole ulospääsyä.

Varsinainen logiikka on toteutettu luokassa Movement.Logic joka toteuttaa metodit Forward(), Backtrack() sekä metodit joilla tarkistetaan onko vasemmalla tai oikealla polkuja.

Tarkistusmetodit toimivat siten että robottia käännetään 90 astetta tiettyyn suuntaan ja tarkistetaan onko alle 20 sentin päässä seinää. Jos seinä on olemassa, palauttaa metodi False ja mikäli seinää ei ole, palautetaan True.

## Testaus

Testitapaus 1:

Tässä testissä Laby aloittaa ruudusta jonka jokaisella puolella on seinä.

Testi onnistui. Laby tekee useita "turhia" mittauksia mutta tajuaa kuitenkin olevansa jumissa ja lopettaa ohjelmansa suorittamisen.

Seinä	Seinä	Seinä
Seinä	← Laby	Seinä
Seinä	Seinä	Seinä

Testitapaus 2:

Laby aloittaa labyrintissa joka on testitapaus 1:n kaltainen paitsi että siinä on yksi pystyrivi lisää ja liikkumatilaa onkin 2 ruutua

Seinä	Seinä	Seinä	Seinä
Seinä	← Laby		Seinä
Seinä	Seinä	Seinä	Seinä

Tämä testitapaus epäonnistui ensimmäisellä kerralla, jonka jälkeen jouduin lisäämään lisää erikoistapaus tarkistuksia. Tässäkin Laby tekee jonkin verran turhia mittauksia mutta onnistuu silti ymmärtämään ettei ulospääsyä olemassa

Testitapaus 3:

Seuraavanlainen labyrintti:

Seinä	Seinä	Seinä	19	Seinä	Seinä	Seinä	
Seinä	← Laby	Seinä	16 / 18	15	14	Seinä	
Seinä	1	Seinä	17	Seinä	13	12	Seinä
Seinä	2	Seinä	Seinä	7	Seinä	11	Seinä
Seinä	3	4	5	6 / 8	9	10	
Seinä	Seinä	Seinä	Seinä	Seinä	Seinä	Seinä	

Labyrinttiin numeroin merkattu järjestys jossa se ruudut kulki. Tässä huomataan että Laby ei etsi lyhintä, vaan jonkin reitin mutta käytös on täysin oletettua.  
Testi on siis onnistunut

Testitapaus 4:

muuten sama labyrintti kuin yllä, mutta aiempi maali on tukittu.

Seinä	Seinä	Seinä	19	Seinä	Seinä	Seinä	
Seinä	← Laby	Seinä	16 / 18 / 20	15 / 21	14 / 22	Seinä	
Seinä	1	Seinä	17	Seinä	13 / 23	12 / 24	Seinä
Seinä	2	Seinä	Seinä	7	Seinä	11 / 25	Seinä
Seinä	3	4	5	6 / 8	9	10 / 26	27
Seinä	Seinä	Seinä	Seinä	Seinä	Seinä	Seinä	

Huomioitavaa teistä on se että ruudussa 3,4 (eli järjestyksessä #17) ei käydä kuin kerran. Käyttäytyminen toivottua ja testi onnistunut.

Testitapaus 5:

Seuraavanlainen labyrintti:

Seinä	Seinä	Seinä	Seinä
Seinä	Laby →		Seinä
Seinä			Seinä
Seinä	Seinä	Seinä	Seinä

Testi epäonnistunut: Laby jää pyörimään ympyrää, vaikkakin tämä on odotetun mukaista toimintaa, ei Laby silti suorita tehtävänsä.

## Testitapaus 6:

Seinä	Seinä	Seinä	Seinä
Seinä	2 / 4	1 / 5	← Laby / 6
Seinä	3	Seinä	Seinä
Seinä	Seinä		

Testi onnistui, Laby ajoi umpikujaan, sen jälkeen peruutti alkuun, kääntyi ympäri ja ajoi ulos

## Rajoitukset ja tulevaisuus

Lattian / pöydän tulee olla käytännössä puhdas ja esimerkiksi sähköteippi estää robotin arvattavan käytöksen kokonaan.

Ongelmia aiheuttaa myös moottorien satunnainen epätarkkuus (Satunnaiset internet forumit ehdottavat että heittoa on n.2%) ja koska robotin toiminnallisuus perustuu ainaisiin 90 asteen käänöksiin niin voi nämä pienet vaihtelut aiheuttaa onglemia pidemmissä labyrinteissä.

Etäisyysensorin toimintakaan ei ole täysin luotettavaa. Esimerkiksi alle 5 sentin päässä tehtävät mittaukset saattavat antaa vastaukseksi 2,5 metriä jolloin robotti luulee, että siinä ei ole seinää ja ajaa sitä pään.

Tulevaisuden kehityksessä päällimmäisenä olisi mahdollisesti silmukan tunnistaminen sillä tällä hetkellä robotti voi jäädä silmukkaan jumiin.

## Käyttöohje

Koottuasi robotin, tarkista että renkaiden koko ja etäisyys toisistaan täsmäävät, sen jälkeen aja build.xml tiedoston ja sen jälkeen lataa "Laby.java" robotin muistiin. Tämän jälkeen toimi seuraavasti:

Muista että labyrintin tulee muodostua tasan yhtäsuurista neliöistä ja jokaisen kulman on oltava 90 astetta. Älä myöskään teet silmukoita sillä robotti voi jäädä niihin pyörimään.

- 1) Aseta robotti jonkin "neliön" keskelle ja aseta se osoittamaan haluamaasi suuntaan
- 2) Aja Laby.java
- 3) Katso kun robo ajelee labyrintissä.
- 4) a) Jos labyrintista voi löytää ulos, robotti löytää sinne aikanaan  
b) Mikäli labyrintista ei ole ulospääsyä, robotti tutkii kaikki reitit ja palaa sen jälkeen alkuun ja lopettaa ohjelman suorittamisen.

5. Pidä kosketussensoria pohjassa kunnes robotti boottaa itsensä.