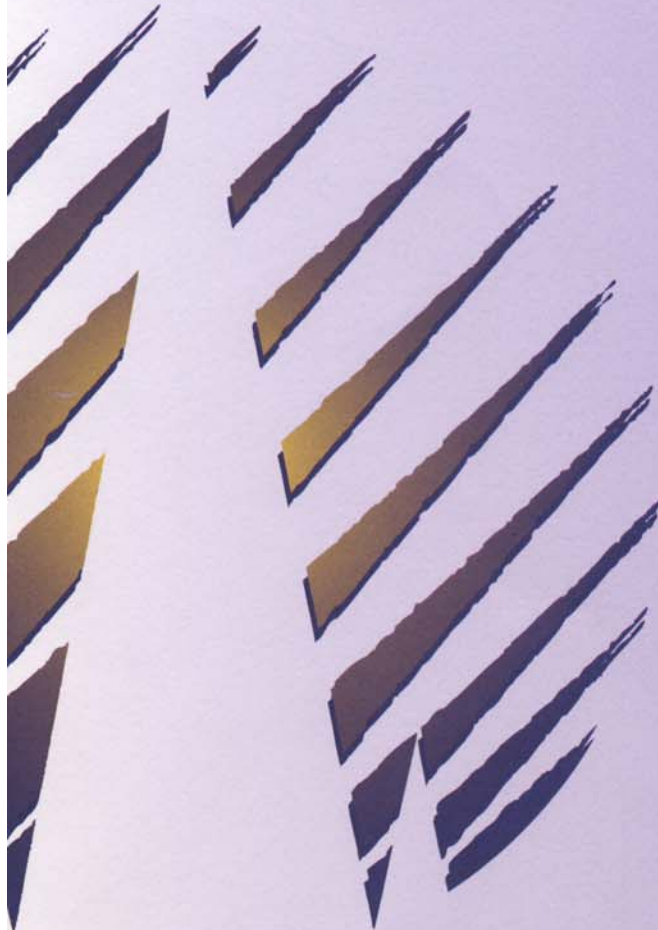


HiST/AITeL - RAPPORT nr 2 2004
Avdeling for informatikk og e-læring
Høgskolen i Sør-Trøndelag



Fagopplegg for bruk av Lego Mindstorm Robolab og objektorientert programmering i et Prosjektfag



SKRIFTSERIER VED HØGSKOLEN I SØR-TRØNDELAG

ISBN 82-7877-101-4

ISSN 1503-0423

HiST/AITeL Rapport nummer 2

**Fagopplegg for bruk av Lego Mindstorm Robolab og
objektorientert programmering i et Prosjektfag.**

av

Grethe Sandstrak og Bjørn Klefstad

grethe.sandstrak@aitel.hist.no

bjorn.klefstad@aitel.hist.no

**Høgskolen i Sør-Trøndelag
Avdeling for informatikk og e-læring
juni 2004**

Sammendrag

Denne rapporten viser hvordan man kan bruke Lego Mindstorm Robolab som case i et prosjektfag, samtidig som det støtter opp om grunnopplæringen i Javaprogrammering. Rapporten ser også nærmere på målsettingen med å ta dette i bruk i undervisningen på Høgskolenivå.

Rapporten har følgende inndeling:

Kapittel 1 omhandler selve legosettet – Lego Mindstorm Robolab, hva finnes det av utstyr og hvilke investeringer bør en foreta for å få gjennomført et slikt kurs.

Kapittel 2 ser vi nærmere på programmeringsspråket LeJOS som er en noe begrenset utgave av Java. Videre kommer det en teknisk beskrivelse av hvordan man installerer nødvendig programvare, og et lite eksempel som demonstrerer prosessen fra et program skrives inn i en editor, til det kan kjøres på en Lego robot.

Kapittel 3 gir en detaljert beskrivelse av fagopplegget som er blitt utarbeidet og gjennomført.

Kapittel 4 inneholder de konklusjonene vi har trukket på bakgrunn av det arbeidet som er gjort. Vi har sett at Prosjektfaget har vist seg som en klar motivasjonskilde for programmeringsfaget. Visualisering av programkoden gjennom adferden til robotene har medvirket til en effektivisering av læringsprosessen og studentene har fått bra utbytte av å arbeide i team samtidig som de har hatt det gøy.

Innhold

Sammendrag.....	3
Forord.....	5
Innledning.....	6
1. Lego Mindstorm Robolab	7
1.1. Innkjøp av utstyr.....	8
2. Hva er LeJOS?	11
2.1. Installasjonsveiledning	12
3. Fagopplegg	14
3.1. Forelesninger med tilhørende leksjoner	14
3.2. Teammøter	15
3.3. Øvingsopplegg	15
3.4. Prosjektoppgave	18
3.5. Evaluering	19
4. Konklusjon	20
5. Ressurser	21

Forord

Denne rapporten fra Avdeling for informatikk og e-læring ved Høgskolen i Sør-Trøndelag, inneholder en praktisk rettet brukerveiledning for å ta i bruk Lego Mindstorm Robolab og LeJOS i prosjektfag.

Vi har gjennomført faget PO 701D Prosjektteknikk i perioden høsten 2003 for 1. kl dataingeniør ved Høgskolen i Sør-Trøndelag ved Avdeling for informatikk og e-læring. I denne rapporten oppsummerer vi våre erfaringer med bruk av Lego Mindstorm Robolab i prosjektundervisning.

Vi har tatt utgangspunkt i at målgruppen for denne rapporten har erfaring med grunnleggende Java-programmering og objektorientering og dermed kjent med de tekniske beskrivelsene og uttrykkene som primært er beregnet på fagfolk.

Vi vil rette en takk til Jarle Larsen som var primus motor i pionerfasen i forbindelse oppstarten av prosjektfaget.

Takk også til Olav Skundberg for verdifulle tilbakemeldinger i forbindelse med rapporten og hjelp med publisering.

Bjørn Klefstad og Grethe Sandstrak

Innledning

Lego Mindstorm Robolab er pr i dag i bruk på flere ulike nivåer innen undervisning. Du finner det i bruk helt i fra barneskolen og oppover til og med Universitet og Høgskoler. Denne artikkelen tar for seg hvordan dette kan tas i bruk på Høgskolenivå. Et naturlig spørsmål som da dukker opp er hva vi ønsker å oppnå med å bruke Lego Mindstorm Robolab. I vårt tilfelle har vi brukt dette med bakgrunn i tre ulike målsettinger:

- Studentene skal få økt forståelse for objektorientert programmering (Java)
- Studenten skal praktisere hvordan man arbeider i Team
- Studenten skal gjennomføre et prosjektarbeid og ha det gøy

Objektorientert programmering oppleves av svært mange studenter som vanskelig. I starten blir det mange nye begreper å forholde seg til og man må holde mange baller i luften samtidig. Man må jobbe en tid med de grunnleggende tingene før man begynner å se sammenhengene og forstår hvordan objektorienteringen er bygd opp. For mange oppleves denne mellomfasen som frustrerende. Og det er nettopp i denne fasen vi kan se nytteverdien av å ta i bruk Lego Mindstorm Robolab. Da får den enkelte student prøvd ut små kodebiter som overføres til Legoroboten og de får prøvd ut programkoden i praksis. På denne måten kan de eksperimentere litt med programkoden og umiddelbart se virkningen av dette i praksis. Vi har laget et øvingsopplegg som bygger videre på den programkoden de allerede har laget i starten. På den måten illustreres nytteverdien av å lage selvstendige generelle kodebiter som vi kan benytte oss av senere. Kravene til objektorientering av programkoden øker utover i kurset. Lego Mindstorm Robolab er derfor godt egnet som støttefag til grunnkurs i Javaprogrammering.

Å arbeide i team er i utgangspunktet en praktisering av Nils Arne Eggens godfotteori. Det gjelder å sette det kollektive i sentrum fremfor å arbeide for å fremme egne interesser. Du skal ta felles ansvar for de områdene der du har spesielle kvaliteter og dermed gjøre de andre god. Først setter man opp et felles mål for teamet og avklarer hvordan de ønsker at samarbeidet skal gjennomføres (arbeidskontrakt). Deretter bidrar alle teammedlemmene på best mulig måte, slik at teamet i fellesskap når det oppsatte målet. For å få prøvd ut dette i praksis må studentene løse en felles oppgave, som er av en slik karakter at det krever at de samarbeider. Og det er her Lego Mindstorm Robolab kommer inn i bildet. For å kunne løse oppgavene kreves det at teamet er kreative i sin utforming av løsning, behersker legobygging og at de behersker de tekniske utfordringene med å lage et program og overføre dette til Legoroboten. I praksis vil vi ganske raskt se at enkelte teammedlemmer bidrar mest på den kreative siden, mens andre er kløppere til å bygge Lego, mens atter andre igjen har sin styrke på programmeringssiden. Bruken av Lego Mindstorm Robolab er derfor godt egnet for å praktisere arbeid i team.

I de aller fleste prosjektarbeid er det fokus på dokumentasjonen av hva som er gjennomført. Dette betyr at studentene må produsere store mengder rapporter og føler at deres kreativitet kveles av mengden rapporter. Selve oppgaven som skulle løses var ikke all verden og gjennomføringen gikk greit, men så oppleves produksjonen av alle rapportene som nærmest altoverskyggende. Ved bruk av Lego Mindstorm Robolab får man prøvd ut om det endelige produktet fungerer i praksis. Man kan forholde seg til et konkret resultat, en selvbygd legorobot. Og når denne testes ser man raskt om resultatet fungerer eller ikke. Studenten opplever det som svært positivt å forholde seg til et konkret resultat og at de får konsentrert

seg om å løse oppgaven istedenfor å produsere mange rapporter. Lego Mindstorm Robolab gir også studentene utfordringer på mange ulike plan og de har det gøy underveis. Dette avspeiles blant annet i at de som regel bruker langt mer tid enn forventet på å løse de ulike oppgavene. De bygger, programmerer, tester, gjør endringer, tester igjen, osv. til de kommer frem til et tilfredsstillende resultat. Dette bidrar igjen til at studentene generelt får et mye mer positivt forhold til programmering. Bruken av Lego Mindstorm Robolab vekker også konkurranseinstinktet hos studentene og de legger mye vekt på å lage en bedre løsning enn de andre teamene.

1. Lego Mindstorm Robolab

Lego Mindstorm Robolab er en serie fra LEGO som brukes til å utvikle og programmere roboter. Den sentrale enheten i Robolab er RCXen (Robotics Command Explorer):



Figur 1-1 RCX - enheten

Lik en vanlig datamaskin har Lego's RCX en prosessor (Hitachi H8/3292 – 16 mhz). Videre har RCX en IR (infrarød) sender og mottager, en LCD - miniskjerm (1 – 3 linjer), en høyttaler, 3 motor kontroll enheter (9 volt) (A, B, C), 3 sensor mottaker enheter (1, 2, 3), 16 Kb ROM – hukommelse (Read Only Memory) og 32 Kb RAM hukommelse (Random Acces Memory) hvor man kan skrive egne programmer.

RCXen er bygd opp med hardware som nederste lag, det neste laget er ROM – hukommelse. ROM – inneholder programmer som gjør det mulig for RCXens maskinvare å fungere. ROM tilsvarer BIOSen på en vanlig PC. Over dette laget finner vi RAM med to funksjoner: Først ligger *Firmware*- programmet som vi kan si er det samme som operativsystemet (dos, XP) på en vanlig PC.

Firmware har som oppgave å oversette dine programmer til maskinkode som RCXen forstår. Javaprogrammer laget for PC er oversatt til bytekod gjennom kompilering som er mer kompakt enn Javakoden. For at denne koden skal være forståelig for RCXen må den oversettes av *Firmware*en til maskinkode.

RCX kan ikke sammenlignes med minnekapasiteten til vanlige datamaskiner:

- ROM 16 Kb

- RAM 32 Kb
 - Firmware ca 16 Kb,
 - brukerprogram opptil 6 Kb
 - tolke kode og behandle data under kjøring av programmet ca 10 kb.

RCX:

RAM 32 kb	Ditt program
	LeJOS klasser
	LeJOS Firmware 16 kb
ROM 16 kb	
Maskinvare	

Med Robolab – pakken følger det et grafisk, hendelsesstyrt programmeringsverktøy. Brukeren trenger ikke skrive kode og det kreves ingen programmeringskunnskaper for å ta dette i bruk. Bakdelen med verktøyet er at man ikke får utnyttet alle mulighetene som ligger i RCX - enheten. Fordelen er at man som uerfaren programmerer får en grafisk framstilling av hvordan et program er bygd opp med moduler, hendelser, kontrollstrukturer og variabler.

I dette kurset vil vi ikke bruke dette programmeringsverktøyet og starter isteden direkte med å programmere Robotene med Javakode. For å få til dette trenger vi LeJOS. LeJOS (Lego Java Operating System) er et programmeringsspråk som kan styre Lego's programmerbare RCX prosessor. Mer om LeJOS i kapittel 3.

Før vi går dypere inn i LeJOS må vi se nærmere på hva som finnes av utstyr og hva som trengs for å komme i gang.

1.1. Innkjøp av utstyr

Hva finnes av utstyr? Når det gjelder Lego, så finnes det mye forskjellig på markedet som kan benyttes i et slikt kurs. Man kan kjøpe inn "startpakker" som inneholder det man trenger for å komme i gang, oppgraderingspakker, enkeltdeler og spesialdelar.

Startpakker

Selve Robolab startsett inneholder 1 RCX, 2 motorer, to trykksensorer, en lyssensor og mange ulike teknikk-legoklosser som man kan utvikle ulike roboter av.



Figur 1-2 Lego Mindstorm Robolab

Startpakken inneholder følgende:

- En grønn byggekasse som inneholder 828 byggeelementer (teknikk-lego), blant annet en RCX og ett IR-tårn. I tillegg ligger det 2 motorer, 2 trykksensorer, 2 lyssensorer og mye annet. Alt du trenger for å komme i gang, følger med.
- Det medfølger også en byggebok på 88 sider med tips til ulike halvferdige modeller.
- CD-rom med selve ROBOLAB-programmet i norsk og engelsk versjon.

Dette Robolab settet inneholder bare skisser til enkelte løsninger slik at brukeren selv må fullføre de ulike prosjektene. Dette kalles ofte for "open ended" som betyr at det ikke er gitt noen eksakt oppskrift på å hvordan man når målene.

Lyssensor:



Lyssensor avleser lysmengder mellom 0,6 og 760 Lux. Benyttes på modeller som skal reagere på endringer i lysforholdene. Ved dataoppsamling kan denne sensor inngå i mange forsøk.

Trykksensor:



Trykksensor registrerer inn (0) og ut (1). Benyttes til å aktivere motorer eller lamper og kan også brukes til å telle personer, biler eller andre ting.

Oppgraderingspakker

Her kan vi velge mellom flere ulike typer oppgraderingspakker. Det finnes ressurskasser som for eksempel "Ressurskasse Teknologi" som inneholder 1100 deler: byggeelementer, tannhjul, akslinger osv. Det finnes også temaesker som for eksempel "Det intelligente huset" som er et spennende undervisningsopplegg omkring moderne innretninger i hjemmene:

automatiske garasjeportåpnere, utelys, nøkkelkort, tyveri og brannalarmer, termostater osv. Dette settet skiller seg fra Robolab settet ved at det finnes detaljerte byggetegninger.



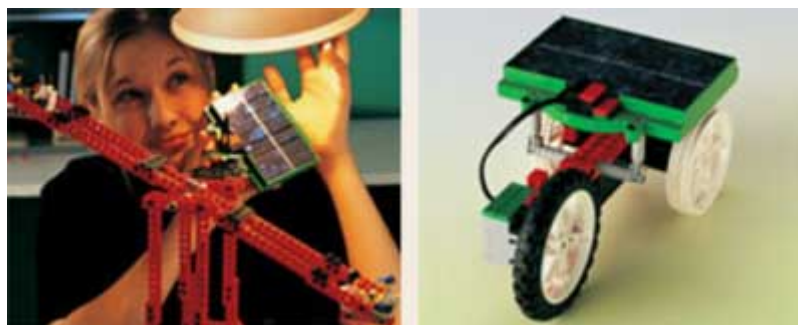
Figur 1-3 Oppgraderings pakke

Ellers kan alle typer standard legoklosser brukes når man bygger robotene.

Ekstradeler

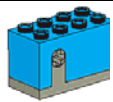
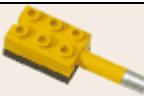
Videre finnes det ekstrautstyr som man kan supplere med. Det som er mest interessant er nok de ulike sensorene man kan få tak i. I tillegg til de som følger med i startsettet (lyssensor og trykksensor) så finnes det temperatursensorer, vinkel-/ rotasjonssensor, lufttrykksensor, luftfuktighetssensor, PH-sensor, Lydsensor, Spenningssensor. Man kan sågar også lage egne sensorer og ved hjelp av en digital/analogt adapter ta i bruk disse.

Andre morsomme deler som finnes på markedet er trykktank, sylindrer med pumpe og slange, pneumatisk veksler, solcellepanel og kondensator.



Spesialdeler

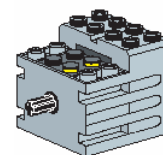
Spesialdeler som ikke følger med startpakkene, som for eksempel temperaturmåler, elektronisk fartsmåler, rotasjonssensorer, vinkelsensorer bør også være en del av utstyrs pakken slik at studentene står friere i forhold til å utvikle spennende og interessante prosjektoppgaver.

Vinkelsensor:		Vinkelsensor registrerer endringer. Den kan programmeres til å reagere på enten rotasjoner eller vinkelgrader. Den deler en omdreining (360) opp i 1/16. Ved datainnsamling kan den bl.a. benyttes til å finne hastigheter, antall rotasjoner mm.
Temperatursensor:		Temperatursensor har et måleområde mellom -20 C og +50 C.

Hva trenger vi?

For å gjennomføre et slikt kurs som skissert i denne artikkelen trenger man en startpakke pr. team. (Hvert team består av fire personer). Videre har vi også investert i en oppgraderingspakke som inneholder ulike spesialklosser og et veiledningshefte for hvordan man kan bygge for eksempel støtfangere. Dette har vist seg å være nyttig i forhold til de studentene som ikke har bygget så mye Lego før. Det kan være greit å støtte seg til et veiledningshefte i startfasen.

Vanlige legoklosser holder som kjent i årevis, men når det gjelder spesialdeler som for eksempel motorer så har disse en svakhet som gjør at de etter noen sesongers bruk kan gå i stykker. Dette gjelder også selve RCXen som inneholder en del elektronikk og dermed er mer sårbar enn de vanlige klossene. Spesielt ved bygging av biler til AITeL-rallyet blir det en del utforkjøringer i utprøvingsfasen.



Ekstra motorer og RCXer er også nyttig å ha når studentene skal jobbe med den egendefinerte prosjektoppgaven. Vi har hatt flere tilfeller der for eksempel studentene trenger mer enn de to motorene som følger med startpakken de har fått utlevert, for å kunne realisere sine prosjektoppgaveforslag. Prosjektoppgaver som inkluderer kommunikasjon mellom to RCXer kan også gjennomføres når man har noen ekstra RCXer på lager.

Dessuten trenger man en del spesialutstyr slik at de ulike prosjektgruppene kan realisere sine planer for prosjektoppgaver. Dette betyr at man kjøper inn ett eller to sett, av ulike typer spesialutstyr. På denne måten får de ulike teamene ulike muligheter i forbindelse med prosjektet. Selv med et relativt beskjedent antall forskjellige typer spesialutstyr, er det nesten bare fantasien som setter grenser for hva de kan bygge og programmere i prosjektet.

2. Hva er LeJOS?

En målsetting med kurset er å gi studentene økt forståelse for programmeringsspråket Java og objektorientering. For å få til dette skal robotene programmeres med Javakode.

LeJOS – Lego Java Operating System – er et programmeringsspråk som kan styre Legos programmerbare RCX prosessor.

LeJOS kan sammenlignes med Java, de bruker begge javac – kompilatoren for å oversette javakoden til bytekode og begge inneholder JVM (Java Virtual Machine) som tar linje for

linje og oversetter til maskinkode. JVM tilsvarer RCX- firmware. Dette programmet må overføres til RCX - enheten fra datamaskinen før vi kan overføre egne javaprogrammer. Gjør vi ikke det, vil ikke RCXen kunne oversette bytekoden til forståelig maskinkode.

JVM (firmware) bruker 16 kb av ram og da er det igjen ca 6 kb til programkoden. Dette er ikke særlig mye plass, men likevel så åpner LeJOS for mange muligheter.

Når man bruker LeJOS må man importere en pakke: **josx.platform.rcx**. I denne pakken får man adgang til metoder som kan kontrollere motorene, LCD – skjermen, motta informasjon fra sensorene og mulighet for å omprogrammere knappene på RCX - enheten. Man kan også programmere tid, strømstyrke, høyttaler og IR – kommunikasjon i LeJOS. Mulighetene er mange. Denne pakken er en del av LeJOS RCX Java Api (Application Program Interface),

Selv om LeJOS bruker samme kompilator som Java så er det noen begrensinger i forhold til Java. De viktigste er:

- LeJOS har ikke Garbage Collection
- switch – setning mangler
- aritmetiske og logiske operasjoner på variabler av type long går ikke, men du kan caste en int til long.
- Maks lengde på tabeller er 511 elementer
-

2.1. Installasjonsveiledning

Hvordan komme i gang med programmering av robotene:

For å kjøre *vanlige* Javaprogrammer gjør vi følgende:

1. Installere JDK på datamaskinen og sette nødvendige miljøvariabler (Path, Classpath)
2. Skriv et Javaprogram
3. Kompilere Javaprogrammet til bytekode ved hjelp av kommandoen `javac.exe`
4. Oversett bytekode til cpu-instruksjoner (maskinkode) avhengig av hvilken maskin programmet skal kjøres på ved hjelp av kommandoen `java.exe`.

Den samme oppskriften må utvides litt når vi skal programmere RCXen. Merk at vi først må installere Java på datamaskinen:

1. Installer JDK og LeJOS på datamaskinen og sett nødvendige miljøvariabler (path, classpath og RCXTTY)
Siste versjon av Java lastes ned fra Sun sine hjemmesider (<http://java.sun.com/>) og siste versjon av LeJOS kan lastes ned fra LeJOS sine sider (<http://lejos.sourceforge.net/>).
2. Overfør LeJOS Firmware (JVM) til RCXen. – `leJOSfirmdl.exe`

3. Skriv et Javaprogram i en passende editor, for eksempel TextPad.
4. Kompilere Javaprogrammet til bytekode – `lejosc.exe`
5. Oversett bytekode til cpu-instruksjoner (maskinkode) avhengig av hvilken maskin programmet skal kjøres på og overføre bytekoden fra datamaskinen til RCXen – `lejos.exe`

Sette miljøvariabler

Følgende miljøvariabler må være satt: Path, Classpath og RCXTTY.

I kurset som vi kjørte var programvaren og miljøvariablene lagt inn av it-ansvarlig på de stasjonære datamaskinene på huset. Flere studenter hadde egne bærbare hvor de foretok nødvendig installasjoner selv.

2.1.1. LeJOS via TextPad

For å skrive programmene kan en vanlig teksteditor benyttes. Vi har god erfaring med å bruke TextPad i Javaprogrammeringsfaget og har valgt å konfigurere denne editoren til også å kunne kompilere LeJOS kode. Dette for å gi studentene et så likt utviklingsmiljø i forhold til Javaprogrammeringsfaget som mulig.

Alternativt til å bruke en editor lik TextPad er å kjøre Java- og LeJOSkommandoer i MSdos-ledetekst hvor mer kryptiske tekstkommandoer skrives for å få kompilert og overført programvaren til RCXen.



RCX



IR - Tårn



Infrarød sender

Oppkobling mot PC



3. Fagopplegg

I dette faget skal studentene utføre et prosjekt. Det arbeides i team (grupper) bestående av 4-5 studenter.

Fagmål:

- Studentene skal lære å samarbeide om en konkret oppgave (Lego Mindstorm Robolab).
- Studenten skal praktisere hvordan man arbeider i Team
- Studenten skal gjennomføre et prosjektarbeid og ha det gøy
- Studentene skal få økt forståelse for objektorientert programmering (Java)

Fagopplegget for kurset er todelt, den første delen tar sikte på å gi studentene en innføring i teamteori, Java programmering og å bli kjent med de ulike delene av legosettet (motorer, trykksensorer, lyssensorer osv) (se Figur 3-1), Del to vies til en selvvalgt prosjektoppgave.



Figur 3-1 Lego Mindstorm Robolab

3.1. Forelesninger med tilhørende leksjoner

I og med at dette er et prosjektfag gis det ikke mange forelesninger, i stedet fordeler man lærerressursene mellom enkelte forelesninger, veiledning på labb og teammøter. Noen emner blir forelest og til hvert emne er det utviklet en leksjon som dekker det aktuelle emnet.

Emner som blir forelest i er:

- Teamteori
- Introduksjon til LeJOS
- LeJOS og objektorientert programmering
- Prosjekthåndbok

Kurset har ingen lærebok og dermed blir utarbeidede leksjoner en viktig del av studentenes læringsprosess. I tillegg brukes Internett flittig som informasjonskilde i forhold til lego og Javaprogrammering.

Forelesningenes form varierer alt etter hvilke emner som gjennomgås. Under emnet teamteori, aktiveres studentene innimellom forelesers ”teori-drypp” med små gruppeoppgaver for å illustrere de ulike teoriene.

Ved gjennomgang av LeJOS og objektorientert programmering, brukes AV-utstyr aktivt. Hele prosessen demonstreres visuelt fra programkoden skrives inn i en editor på en PC til den er lastet ned og kjørt på en Lego robot.

Det siste emnet som foreleses er Prosjekthåndbok. Prosjekthåndboka viser hvilke dokumenter som kreves i forbindelse med prosjektet. Forelesningen kjøres for å gi studentene nødvendig informasjon i forhold til kursets siste halvdel – prosjektgjennomføring.

I tillegg vil det blir gjennomgått løsningsforslag til noen av øvingene. Og om det er noen emner flere har problemer med (for eksempel en og samme programmeringsfeil går igjen hos flere av temaene i løsningen av en oppgave) så kan dette også utdypes/oppklares i disse timene.

3.2. Teammøter

Teammøtene er formelle møter som inkluderer alle teammedlemmene og faglærer. Det gjennomføres fire teammøter pr semester. Ekstraordinære teammøter kan avholdes ved behov.

Det enkelte team har ansvar for å skrive møteinnkalling, møtereferat og lede møtene. De ulike rollene roterer innad i gruppen slik at alle teamets medlemmer får prøvd seg som møteleder og referent.

Teammøtene dekker noen faste poster:

- Teamsamarbeid/ teamdynamikk – hvordan går det
- Demonstrasjon og gjennomgang av øvingsarbeid, er oppgaven løst tilfredsstillende innenfor gitt tidsfrister.
- Deltakelse/ arbeidsfordeling i øvings- prosjektarbeid – Har alle teamets medlemmer bidratt i arbeidet (evt. hvorfor ikke).
- Oppmøte – Teammøtene er obligatoriske.

I tillegg står det fritt til studenter og faglærer å legge til nye poster.

3.3. Øvingsopplegg

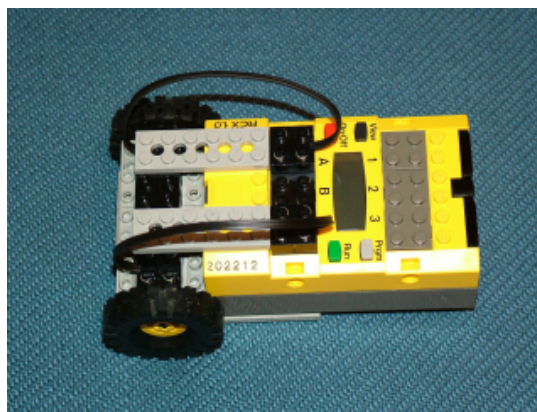
Med bakgrunn i en studentgruppe fra 1.klasse ingeniør har vi utarbeidet et øvingsopplegg som er tilpasset de faglige forutsetninger som denne studentgruppen (1 kl. dataingeniør) innehar.

Det vil si at progresjonen i øvingsopplegget er nøye knyttet opp mot faget Grunnleggende programmering i Java som går parallelt.

Øvingene i går fra å være helt lærerstyrte, der studentene gis detaljerte instruksjoner for hvordan oppgavene skal løses, til gradvis å bli mer studentstyrte med større rom for egne løsninger og kreativitet.

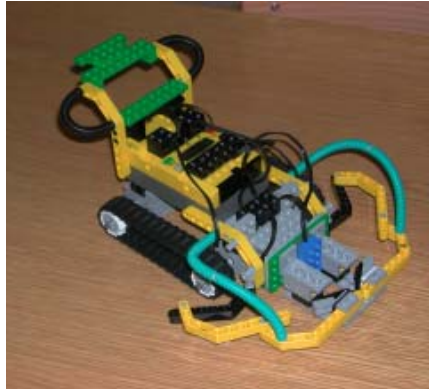
Det gis 5 øvinger totalt hvor de fire første går hver uke, den siste øvingen får studentene noe lengre tid på og er siste utfordring før det skal velges en egnet prosjektoppgave.

- I øving 1 skal studentene bli bedre kjent med hverandre og blant annet sette opp en arbeidskontrakt som viser hvilke regler som gjelder for samarbeidet. Eventuelle konflikter som skulle oppstå senere skal løses med utgangspunkt i innholdet i arbeidskontrakten. Arbeidet som gjøres her er viktig for teamets videre utvikling.
- I øving 2 skal studentene gjøre seg kjent med Legosettet, bygge seg sin egen bil (en enkel bil - Figur 3-2) etter gitte spesifikasjoner. Videre skal de lage et enkelt Javaprogram som får bilen til å kjøre i et kvadratisk mønster, skrive tekst til displayet, samt å pipe når den er ferdig. Studentene har på dette tidspunktet laget sine første Javaprogram i Javakurset og er dermed kjent med utviklingsmiljøet for Java. De er dermed kjent med bruk av editor (vi bruker TextPad) og har vært gjennom noen runder med kompilering og kjøring av Javaprogrammer. I denne øvingen konfigureres TextPad til også å utføre LeJOS-kommandoer slik at samme utviklingsmiljø kan benyttes.



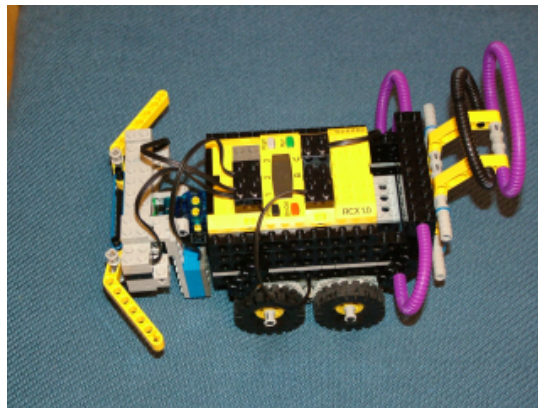
Figur 3-2 Øving 2 - enkel robotbil.

- I øving 3 skal studentene bli bedre kjent med klassebegrepet i Java og noen ferdigdefinerte LeJOS – klasser. Videre skal bilen fra øving 2 utvides med en solid støtfanger som bruker trykksensorer for å registrere at bilen støter borti et eller annet. Dersom den støter borti noe må den rygge og svinge litt før den igjen kjører fremover. Det stilles krav til man bruker klasser og at koden skal være objektorientert. (se Figur 3-3)



Figur 3-3 Øving 3 - robot med solid støtfanger

- I øving 4 skal studentene ytterligere videreutvikle bilen med lyssensor (se Figur 3-4). Lyssensoren (e) skal kunne skille mellom lyse og mørke områder. Når sensoren merker en slik overgang skal roboten gjøre et eller annet. Samtidig stilles det nå større krav til at programkoden er objektorientert. Dette betyr at studenten både må lage egne klasser, opprette objekter og kunne bruke ferdigdefinerte klasser.



Figur 3-4 Øving 4 - robot med støtfanger og lyssensor

- I 5. og siste øving før prosjektet skal studentene bruke det de har lært til nå og bygge en egnet bil som skal delta i et rallyløp. Dette løpet foregår på en bane satt sammen av flere bord i et slags sikk-sakk mønster (Se Figur 3-5).



Figur 3-5 Utforming av Rally-banen

- Rallybanen kan i prinsippet endres slik man selv ønsker, men bør være fastlagt før studenten setter i gang med å løse oppgaven. Den som raskest mulig forserer hele banen fra start til mål har vunnet. Løpet kjøres med tidtaking der en og en bil starter i gangen. Hvert team får to muligheter til å fullføre løpet og det kjøres to heat. (Se Figur 3-6).



Figur 3-6 Rally-løp, Roboten har tre lyssensorer som registrerer om den er over bordet eller gulvet.

3.4. Prosjektoppgave

Del to av kurset inneholder en selvvalgt prosjektoppgave. Ut fra de erfaringer og det studentene har lært i del 1 av kurset, skal de velge seg en prosjektoppgave med utgangspunkt i de lego-delene hvert team disponerer. Krav som stilles til oppgaven er at den skal være løsbar, av passende størrelse og det skal skrives prosjektdokumentasjon i form av en prosjekthåndbok. Før studentene setter i gang skal oppgavevalg godkjennes av faglærer.

Prosjekthåndbok skal inneholde følgende:

- Arbeidskontrakt
- Oppgavebeskrivelse
- Uml diagrammer (klassediagram og aktivitetsdiagram)
- Egenevaluering av prosjektet
- Timelister, med statusrapport
- Møteinnkalling/ Møtereferat

Vi har sett svært mange ulike varianter av legoroboter som resultat av prosjektet. For eksempel traverskran, graftegner, kalkulator som skriver ut resultatet på papir, drinkmikser, nonstop sorterer, Marsmallowsgriller, båt, brusboksoppsamler, tåknuser, danserobot som kommuniserer med en annen robot samtidig som de veksler på å spille musikk. Så her er det nesten bare fantasien og kreativiteten som setter grenser.

Ved endt prosjekt skal alle teamene holde en presentasjon for veiledere og resten av klassen, hvor de viser fram roboten de har utviklet, samtidig som de redegjør for selve prosessen – hvordan har samarbeidet gått? Har det vært noen problemer underveis? Osv.

3.4.1. Eksempler på prosjektoppgaver som er løst:

Morsetolker (Figur 3-7): Fungere slik at en robot leser papirstrimler med morsekoder ved hjelp av en lyssensor. Morsekodene tolkes av roboten og overføres så via infrarød kommunikasjon til en annen robot. Denne roboten skriver så ut bokstavene på papir.



Figur 3-7 Morsetolker

Katapult (Figur 3-8): Denne konstruksjonen består av en kran som løfter opp en liten ting, for eksempel en flaskekork, før den snur seg 90 grader og legger ned flaskekorken i katapulten på den andre roboten. Når denne operasjonen er utført sendes det et signal fra den første roboten til den andre om å løse ut katapulten. Flaskekorken slenges så med stor kraft ut i rommet.



Figur 3-8 Katapult

3.5. Evaluering

Prosjektfaget evalueres med karakter Bestått/ Ikke bestått. Karakter gis etter følgende kriterier:

- Gjennomført og godkjent øvingsopplegg (25 %)
- Deltakelse og gjennomføring av teammøter med veileder (25 %)
- Gjennomført prosjekt og godkjent prosjektdokumentasjon (50 %)

4. Konklusjon

Ut fra oppsatte fagmål kan vi konkludere med følgende etter gjennomført kurs:

- Studentene har gjennomført en selvvalgt prosjektoppgave som har inneholdt utfordringer både i forhold til programmering, teamsamarbeid, prosjektdokumentasjon og konstruksjon av legoroboter.
- Studentene har ved hjelp av Robolabsettet fått praktisert ulike roller i forbindelse med teamarbeid og på bakgrunn av dette, lært å samarbeide.
- Ut fra studentenes tilbakemeldinger kan vi konkludere at prosjektfaget har vist seg som en motivasjonskilde for programmeringsfaget og dermed inspirert til økt innsats.
- Visualisering av programkoden gjennom adferden til robotene har medvirket til en effektivisering av læringsprosessen og har bidratt til en større forståelse for grunnleggende programmeringsteknikker.
- På bakgrunn av tilbakemeldinger og gjennomførte evalueringer av faget kan vi konkludere med at studentene har hatt det gøy med faget.

5. Ressurser

Prosjektfagets hjemmeside [Online] Tilgjengelig: http://www.aitel.hist.no/fag/_phk1/

I tilknytning til Lego Mindstorm Robolab og leJOS så finnes det mange ressurser på Internett. Under finner dere en samling av nyttige lenker vi har brukt i undervisningen.

Giulio Ferrari [et al.]. 2002: "Programming Lego Mindstorms with Java, The ultimate tool for Mindstorms Maniacs". Rockland, Mass. : Syngress ISBN 1-928994-55-5

Lego produkter LEGO Group, 2003 [Online] Tilgjengelig:
<http://www.lego.com/eng/products/next/>

Lego's side om utdanning. LEGO Group, 2003 [Online] Tilgjengelig:
<http://www.lego.com/dacta/home.asp>

Artikkel om Lego Mindstorm [Online] Tilgjengelig:
http://folk.uio.no/terjekk/foredrag/datalego/artikkel1_Lego.htm

Lejos's side [Online] Tilgjengelig: <http://lejos.sourceforge.net/>

Lejos API [Online] Tilgjengelig: <http://lejos.sourceforge.net/apidocs/index.html>

Java API [Online] Tilgjengelig: <http://java.sun.com/j2se/1.4.2/docs/api/index.html>



HØGSKOLEN
I SØR-TRØNDELAG

Avdeling for helse- og sosialfag
Avdeling for lærerutdanning og tegnspråk
Avdeling for teknologi
Avdeling for informatikk og e-læring
utgir:

HiST-rapport

- avhandlinger
- monografier
- prosjektrapporter

HiST-notat

- artikler
- foredrag, konferanseinnlegg
- notater, delrapporter

Avdeling for økonomisk-administrativ utdanning
utgir:

TØH-serien

Formålet med skriftseriene er å

- stimulere den enkelte til å publisere og dokumentere sitt faglige arbeid
- profilere høgskolens faglige/vitenskapelige arbeid
- dokumentere den faglige/vitenskapelige aktivitet overfor interne og eksterne instanser

*Skriftseriene kan bestilles hos de avdelinger som utgir seriene.
Høgskolen i Sør-Trøndelag, 7004 Trondheim • Telefon 73 55 90 00*