

KTH KISTA

Linefollower

Med parprogrammering i NXC

Simon Boström

2014-09-04

simbos@kth.se

Introduktionskurs i Datateknik

Sammanfattning

Laborationstillfället var till för att man som ny på KTH skulle lära sig enkel programmering. Utöver detta lär man sig även hur man skriver en rapport och hur man navigerar i Bilda och Daisy.

En kod med diverse fel skulle rättas till för att få en legorobot att följa en linje. När roboten sedan kör in i väggen skulle den stanna och visa text på displayen. Detta skulle uppnås i par. Efter lite diskuterande hade detta uppnåtts utan allt för stora problem. Efter denna laboration har det kommit fram att detta är ett bra sätt för nybörjare att lära sig programmera och egenskaper som kan komma att behövas i framtiden förvärfvas.

Innehållsförteckning

1. Inledning	3
1.1 Bakgrund	3
1.2 Syfte och målsättning	3
2. Genomförande	3
3. Resultat	4
4. Analys	4
5. Diskussion	4
Referenser	4
Bilagor	5

1. Inledning

För att komma in i programmeringstänket fick vi i uppgift att felsöka koden till en legorobot som skulle följa en linje på golvet. Koden var skriven i språket NXC, vilket fungerar utmärkt till att både lära sig programmering och friska upp minnet om man så önskar.

1.1 Bakgrund

Att programmera just en legorobot känns kanske inte som något man kan förväntas göra i framtiden, men att få lära sig allt som ingår i att anmäla sig till laborationer, navigera Bilda och Daisy, samt att få en inblick i programmering är en vital del som passar bra för en ny student på KTH.

1.2 Syfte och målsättning

Syftet med uppgiften är att på ett roligt och relativt enkelt sätt introducera programmering i form av parprogrammering.

Syftet är också att introducera arbetsgången vid ingenjörsarbete samt ge träning och arbetsvana i de IT-system för utbildning som finns vid ICT-skolan.

Uppgiften syftar även till att ge träning i felsökning och testning och därmed ge en grund till effektivt programmeringsarbete i framtiden. Förhoppningen är även att studenterna ska få en förståelse för hur små enkla fel i koden kan ge oväntade resultat vid körning, och på så sätt bidra till ett noggrannhetstänk.

2. Genomförande

Jag kom till lokalen där laborationen skulle hållas och blev hopparad med en annan student. Programmet BrixCC laddades ner och installerades, koden laddades ner, drivrutiner installerades och legoroboten undersöktes. Efter att ha kopplat in roboten till datorn och öppnat koden läste jag och min partner igenom koden för att få en överblick av situationen. Vi läste sedan igenom de felmeddelanden vi fick och jobbade med att lösa dem. Efter att ha fått slut på idéer bestämde vi oss för att läsa instruktionsboken. 50+ sidor senare upptäckte vi att där det skulle stå "string" stod det "int". Efter att ha löst detta problem (vilket tog oss lite för lång tid för den triviala naturen hos problemet) testkörde vi roboten. Den betedde sig lite underligt, men när vi fick stopp på den upptäcktes nästa problem: all text som skulle visas på displayen var på samma rad. Efter att vi lokaliserat den del av koden som styrde detta utfördes en enkel matteberäkning och ett minustecken ändrades till ett plustecken, och en hårdkodad siffra ökades.

Vi upptäckte efter att ha löst alla problem med displayen att anledningen till att roboten betedde sig konstigt var att den var felkopplad. Vi ändrade koden istället för att koppla om, då uppgiften inte var att koppla om roboten (och att programmera är betydligt roligare än att koppla). Vi märkte även att två rader kod inte såg bra ut, så vi ordnade detta. Vi satte roboten på den svarta linjen på golvet och körde igång. Den följde linjen bra, men när den skulle svänga fortsatte den rakt fram. Vi kom fram till att det gick för fort så att robotens sensor inte hann uppfatta linjen. Efter att vi sänkt robotens fart fungerade koden perfekt.

Vår parprogrammering var inte av sorten där en person programmerar ett visst antal minuter för att sedan byta, utan vi diskuterade helt enkelt mellan varandra om hur vi bäst skulle lösa problemet. Det kändes smidigare och fungerade bra.

3. Resultat

Radnummer	Ny kod	Kommentar
2	SpeedSlow 30	Den gick för fort.
3	SpeedFast 50	
34	Int -> string	Man kan inte deklarera text som siffror.
35	Simon	Våra namn.
36	Hassan	
44	names -> groupMembers	Snyggare kod (enl. min åsikt).
46	8*i-16 -> 8*i+8	Bättre layout
92&98 94&100	SpeedSlow SpeedFast	Så att roboten kör som den ska.

4. Analys

Det största, och i stort sett enda, problemet låg i att förstå vad koden innebar för att sedan kunna förstå robotens beteende och rätta till felen. Då jag har programmerat innan var det inga större problem bortsett från att komma in i tänket igen och att lära mig ett nytt språk. Men i allmänhet gick allting bra.

5. Diskussion

Jag har i huvudsak lärt mig hur man anmäler sig till grupper och laborationer via Daisy, samt hur man navigerar i Daisy och Bilda. Utöver det så kom jag även, mer eller mindre, tillbaka in i programmeringstänket. Allt detta kommer definitivt att hjälpa i framtiden. Själva språket, NXC, är utmärkt för nybörjare då det är väldigt enkelt att arbeta i, dock ogillade jag det bland annat på grund av layouten.

Medan vårt sätt att programmera (diskutera med varandra istället för att programmera en viss tid var) inte är det mest använda tycker jag att det fungerade bra, särskilt i ett så litet projekt.

Referenser

<https://bilda.kth.se/courseId/11430/content.do?id=22224147>

<https://bilda.kth.se/node.do?id=22300661>

Bilagor

Post Personal note

Introkurs i Datateknik: Robotprogrammeringslabb

Vi skulle programmera en robot så att den lyckades följa en svart linje på golvet och sedan åka in i väggen och skriva gruppdeltagarnas namn på displayen.

Det var en väldigt givande uppgift med många fördelar, till exempel det faktum att det var relativt enkelt och roligt. En nackdel var dock att vi inte hade fått en tillräcklig genomgång av programmet och specifika instruktioner på vad som skulle göras, så mycket tid spenderades med att läsa instruktionsboken.