Proceedings of the Twelfth Belgium-Netherlands AI Conference BNAIC'00, van den Bosch and Weigand (ed.), 2000, pp 29-36

Programming robots is fun: Robocup Jr. 2000

Ben Kröse Rob van den Bogaard Niels Hietbrink

Intelligent Autonomous Systems Group
Department of Computer Science, University of Amsterdam,
Kruislaan 403, 1098 SJ Amsterdam

Abstract

During RoboCup Euro2000 we organized a tournament for schools using LEGO robots: RoboCup Jr. In this paper we present the objectives of RoboCup Jr. and describe the technical aspects of the robots. We also report on the event itself: how it was organized and what sort of tactics and computer programs were developed.

1 Introduction

During the last ten years a number of Dutch Universities started an undergraduate studies in Artificial Intelligence (AI) or modified a specialization program into a full undergraduate program in AI. This means that a first year student can now choose whether he or she wants to study computer science, which is focused on the hard and software of the computing machinery or artificial intelligence, which is focused on the information technology and its use to create intelligent systems or to model them. Apart from the programming skills students are taught subjects such as cognitive science, philosophy, computational linguistics or robotics.

In secondary schools, Computer Science only recently is taught as a subject. Still it is only an optional subject. Computer Science does not have a Central Exam but a School Examination. The students learn how to work with standard computer programs (text processing, internet), learn about the hardware and make a start with the learning of programming and systems design. However, there is still a shortage of good educational material and skilled teachers. In particular there is almost no material with an 'AI flavour' (reasoning, searching, computer vision, robotics). In the field of robotics a number of research groups worldwide has noticed this and started various initiatives. Here we describe our experience with the RoboCup Jr. initiative.

2 Robocup Jr.

During the RoboCup'99 which was held during the IJCAI conference in Stockholm in 1999 the first RoboCup Jr. was arranged by Henrik Lund and Luigi Pagliarini [2]. The aim was to allow children get experience with the programming of robots. A set up using LEGO robots and a simple and clear programming environment was made. This initiative was picked up by the international Robot World Cup Federation (RoboCup) [1]. The aims and structure of RoboCup Jr. can be described as follows.

The aims

The objective of RoboCup Jr. are twofold:

- To contribute to the education of computer science, in particular the information technology which is needed to program autonomous systems. This means that many AI related problems will be incorporated.
- To develop an infrastructure for robot education, and more general the education in programming computers such that they can operate in a real world.

Educational aspects

In [1] several aspects of education in RoboCup Jr. are described:

Learning through design The student has to design robots and software during the project. The modular approach in which both the software and the hardware is provided (in our case LEGO) makes it possible for the students to meet with the artifacts of programming a 'real world system' during the process and to learn from that.

Learning through competition The competitive aspect of the RoboCup Jr. motivates the students to experiment with their programs. Also the test games and qualification games enable the students to analyze the design of the programs (robot behaviour) of their opponents and to apply this knowledge in improving their own programs.

Project oriented education The students learn to work as a team, discussing the possible artifacts and strong points of their program. This again makes the students aware of the power of development of a good program.

3 The LEGO competition

For the RoboCup Jr. 2000 which was held in Amsterdam we used LEGO robots to play soccer games. There were a number of reasons for choosing this material:

- The group of Henrik Lund had developed software for programming LEGO robots to play soccer which was tested at RoboCup '99 in Stockholm

- There is expertise at the University of Amsterdam (Amstel Instituut) to work with LEGO material
- For a reasonable price we were able to have a sufficient number of robots

The robot

The robot is based on a design by Lund of LegoLab Denmark. The basis is a standard Lego Mindstorms set. However, because there are not enough motors and sensors in, separate ordering of missing motors, sensors and parts is required.

RCX The heart of the soccer robot is the LEGO RCX computer. There is a Hitachi H8 RISC processor inside, with 32K of external RAM. The micro-controller is used to control three motors, three sensors, and an infrared serial communications port. An on-chip, 16K ROM contains a driver that is run when the RCX is first powered up. The on-chip driver is extended by downloading 16K of firmware to the RCX. Both the driver and firmware accept and execute commands from the PC through the IR communications port. Additionally, user programs are downloaded to the RCX as byte code and are stored in a 6K region of memory. When instructed to do so, the firmware interprets and executes the byte code of these programs (see [3]).

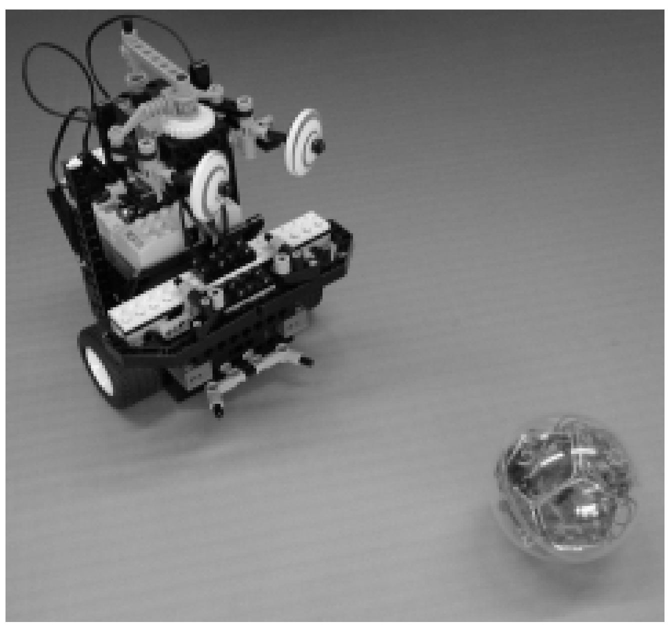

Figure 1: The LEGO robot and the ball with infrared emitters.

Soccer robot The robot which we used was designed especially for the robot soccer game. It is built around an RCX processor. It uses three LEGO motors: two to control the wheels and one to control the eyes. This latter is mainly a 'cosmetic' effect but was also used by the children to monitor the state of the program. There is a variety of sensors mounted n the robot. Two touch sensors are connected to the bumpers to check whether there is a collision left or right. The robot has three light sensors. Two of them are used to detect the light of the ball. A simple 'Braitenberg' principle is used: if the right sensor is giving a higher reading then the robot knows that the ball is to the right. A third light sensor is mounted at the bottom of the robot. In combination with the design of the field (which is a white to black gradient) the robot knows (about) where it is[1].

Figure 2: A fragment of the user interface of the ILF program.

Software The soccer robot is programmed using the 'ILF'-software developed by Lund and Pagliarini which provides children an easy to use programming environment. A soccer robot program consists of a list of behaviours that is cyclically executed by the robot after the program has been downloaded into the RCX and run. Behaviours are selected from a list that is always visible on the left of the screen (see figure 3). Real world examples of the behaviour are available in the form of video clips selectable from the behaviours list. Some behaviours require a parameter to be set: parameter values are selected from a list that pops up whenever a parameter is needed. Behaviours vary from 'Go to the white end', 'Find the ball', 'Go left' to 'Wander randomly' and 'Make a sound'. At the top of the behaviours list are the selection boxes

[1] The program used for the RoboCup Jr. competition reads raw values from the sensors instead of percentages (which is the default). This is to increase the resolution of the light sensors. The values run from 0 to 1024 where 1024 is total darkness and 0 is the maximum amount of light the sensor can register. A touch sensor is a binary sensor, which just outputs 0 (depressed) or 1024 (free). Since the light sensor doesn't reach these values in normal operation, it is possible to connect both a light sensor and a touch sensor to one sensor input. It is then easy to distinguish between the two sensor readings. Since two pairs of touch sensors are combined and connected to the same input channel, it is possible to accomodate all the sensors on the robot (three light sensors and five touch sensors).

of two additional 'permanent' behaviours. The behaviours selected here are continuously executed in parallel with the user program and are used to control whether or not the robot will react to bumper activity and whether the 'eyes' of the robot are pointing in the direction the robot has detected the ball or are just randomly 'looking around'. A push on the download button will load the program and permanent behaviours into the RCX[2].

The field and ball

The field for the soccer game is a grayscale field printed on an oversize A0 (119 x 87 cm), surrounded by a wooden frame of about 13 cm high. The intensity level of the field forms a gradient from white to black, and using a light sensor underneath the robot one can navigate around the field. Before starting the system has to be calibrated. Like in normal soccer there are two goals at opposite ends formed by holes in the wall.

In contrast with the middle-size league in RoboCup, the LEGO robots do not have a camera on-board. This would require too much processing time an would result in a too expensive system. Instead the system uses LEGO light sensors, so this means that an 'active' ball is needed. In RoboCup Jr. 2000 we used the balls which were designed in the LEGOlab of the University of Aarhus (see figure 3). These balls contain LED's which emit infrared light of a wavelength which is optimal for the LEGO sensors. The ball was made out of transparent plastic in which 20 infrared emitters were positioned in a hexagonal grid, so that coverage of all angles is assured. These balls were hand made, but a commercial version will be available soon from a Japanese company Elekit.

The games

There were 12 teams from 8 schools participating in this tournament, in total about 50 children varying from 13 years to 16 years old . In the morning the whole group of participants was given a short general introduction after which every team was assigned to a computer and a robot. We had one supervisor per three groups, which is really needed. The younger children used the 'beginners' mode of the ILF program while the older ones were encouraged to try the intermediate level,

[2]The 'ILF'-software is programmed in Visual Basic and uses the ActiveX (OLE) control interface developed by LEGO. This control (Spirit.ocx) performs the tasks of serial communication with the IR tower, checking and downloading code and loading firmware into the RCX. Most of the functions exported by the control assume that the standard LEGO firmware as shipped with the LEGO Mindstorms set is residing in the RCX. This firmware partitions RCX memory in three main sections. One section contains the memory for 32 16-bit integer variables, the second section is filled with user code and the third is used for datalog storage (sensor/timer/variable readings can be recorded and uploaded to the computer via the IR tower). User code can take the form of tasks or subroutines: Tasks can invoke subroutines and can run in parallel. The RCX can assign tasks to up to 5 programs. Every program can hold 10 tasks and every task can use 8 subroutines. The first task (task 0) of the selected program (the RCX has a program select button) is executed when the green 'run' button on the RCX is pressed. The 'ILF'-software uses concurrent tasks to implement the permanent behaviours described above and the buffered processing of light sensor readings for ball position estimation.

```
Der Manschaft
31 mei 2000 14:58u
a.ilf

SCORE IN WHITE

Robot reacts to Bumpers Activity
Robot Eyes Look at the Ball
---
Beh:1   Vind de Bal
Beh:2   Maak een Geluid: PIEP PIEP
Beh:3   Zoek en Ga Naar Bal: 5 sec.
Beh:4   If Robot sees the Ball Close
Beh:5   Ga Lichter: 6 sec.
Beh:6   END If Robot sees the Ball Close
Beh:7   If Robot is on White Side
Beh:8   Ga Donkerder: 5 sec.
Beh:9   END If White Side
---
```

Figure 3: The program of the winning team. This team used the intermediate level in which it is possible using IF statements

in which they could also use conditionals in their program. Despite the initial fear of programming, after 15 minutes everybody was very enthousiastic. The first tests were carried out to investigate the effect of the parameters of the program on the behaviour of the robot. After one hour we started playing the qualification rounds: from the 12 teams only 8 could be admitted in the quarter finals. We played on four fields simultaniously. After these qualification games we still had teams of young children as well as older ones in the tournament.

After the qualification games there was a lunch break and a demonstration of the 'real' RoboCup games. We noted that some children wanted to continue programming to improve their chances on winning. However, to have a fair tournament (and to have time to recharge the balls) we imposed a strict one hour break. After lunch we played the quarter finals, semi finals and finals on a single field in front of the supporters.

A LEGO Mindstorms set was given as a first price to the winners (Montessori Lyceum Amsterdam). The second and third place also received a price. To our surprise the second price winner was a group with a very simple program. The strength of this program however is the speed it loops such that it is always faster near the ball than complex programs. The winning team had a program which was developed in the intermediate level mode (see figure 3).

```
Esprit 2
31 mei 2000 11:26u
wit.ilf

SCORE IN WHITE

Robot reacts to Bumpers Activity
Robot Eyes Look at the Ball
---
Beh:1   Robotogen Kijken Naar de Bal
Beh:2   Ga Donkerder: 6 sec.
Beh:3   Zoek en Ga Naar Bal: 10 sec.
Beh:4   Vind de Bal
Beh:5   Ga Lichter: 4 sec.
---
```

Figure 4: Example of a program from the beginners level.

4 Conclusions

The RoboCup Jr. games with the LEGO robots provide a very good platform for the AI education of children in the age between 8 and 15. The fact that a program has to be designed which controls a real system, and that this system has to compete with other systems (of which at the moment the physical capabilities are identical) motivates the students enormously.

At the moment we are making an inventory of other systems which have the same objectives. There is an increasing interest in this field of *edutainment*, both from industry as well as from academia. We believe that the AI community will benefit from a good explanation of the problems and education in secondary schools.

References

[1] Hiroaki Kitano, Sho'ji Suzuki and Junichi Akita. RoboCup Jr.: RoboCup for Edutainment. In *Proceedings of Proc. of Int. Conf. On Robotics and Automation 2000, IEEE Press, NJ, 2000*, 2000.

[2] H. H. Lund and L. Pagliarini. RoboCup Jr. with LEGO MINDSTORMS. In *Proceedings of Proc. of Int. Conf. On Robotics and Automation 2000, IEEE Press, NJ, 2000*, 2000.

[3] Kekoa Proudfoot. RCX internals, http://graphics.stanford.edu/ kekoa/rcx/, 1998.

Figure 5: Impressions from the qualification games