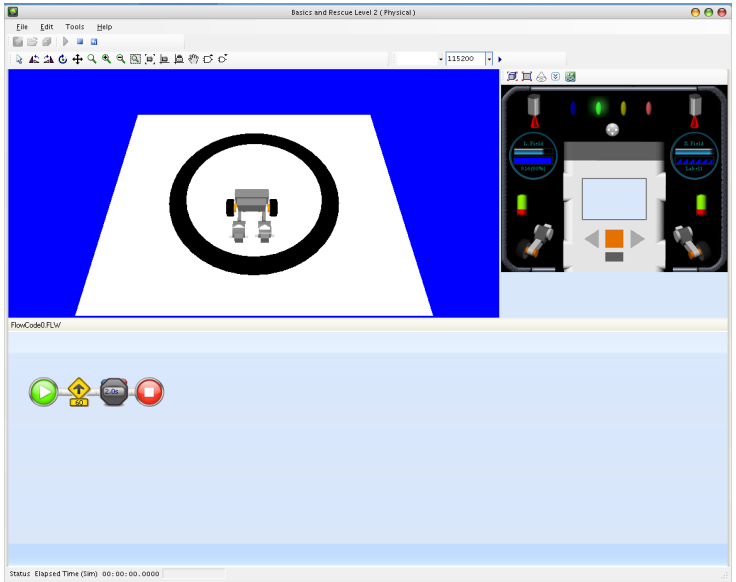
1. Moving Forward

Make the robot drive forward for a short while

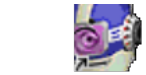
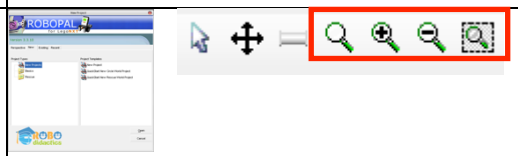
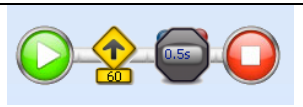
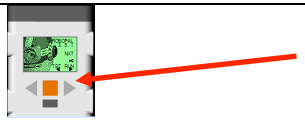
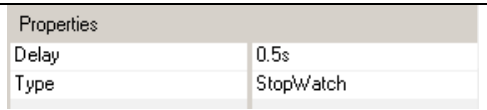


The simple program on the left shows you how to make a robot move forward for two seconds. You will make this program a little more interesting with each assignment. On the back of the instruction cards, you will find step-by-step instructions on how to develop and run this program. The idea is to make the robot stop at a black line.

The first icon starts the program. The second icon – the 'Driver' - starts the motors and the stopwatch next to it then counts two seconds. The last icon stops the program. Try your program out on the simulator: it will show you what your robot can do.

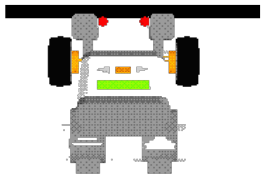
You can use your mouse to move the robot and the left mouse button to make it turn. The blue button on the left-hand side in the menu bar is used to return to the program editor, while the blue button with the little arrow is used to restart the program while still in the simulator.

Do not spend too much time on this program, there is much more to discover. Follow the instructions on the backside of this instruction card and learn how to develop and test your first program.

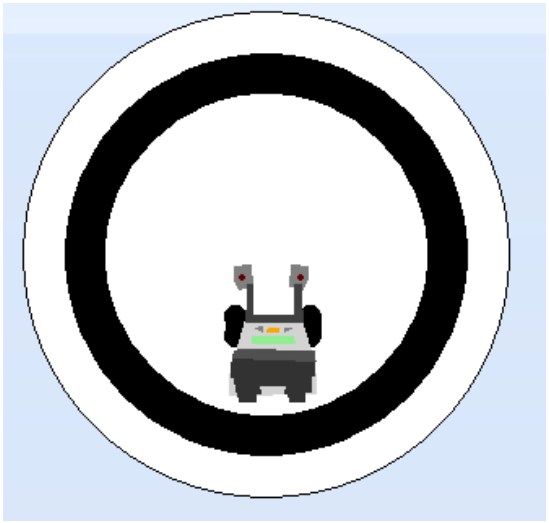
1	 RoboPAL4NXT.lnk	If the RoboPAL program is not running, double click on the RoboPAL icon on the desktop; otherwise go directly to step 2.
2		If you cannot see the selection screen, select it from the menu: File New. Then choose 'Quickstart new Circle World project' and open the project. Click on the bottom panel with the green and red icon. This is the FlowCodeSheet that you will use to make your program. On computers with a small screen, the red icon is sometimes hidden underneath the panel on the right. Use the magnifying glass to make the icons smaller or larger.
3		Select and open the "Driver Controls" from the left-hand column (the ToolBox). Then use the left mouse button to select the 'DriveStraightAhead' icon. This icon is now "attached" to your mouse: move it to the FlowCodeSheet and you will see the icon appear there.
4		Place it next to the green start icon, then press the left mouse button again to release the icon. Once you have placed an icon, you can move it around by using the cursor with the four arrows.
5		Select the stopwatch from 'Program Flow' and place it in the FlowCodeSheet next to the stopwatch icon.
6		Use the cursor with four arrows to move the red Stop icon next to the stopwatch. Your first program is now ready.
7		Use the cursor to click on the upper part of the RobotWorldsSheet: the green arrow of the Run button is now enabled. Click on it to start the simulator.
8		You will see a picture of the NXT robot on the right-hand side of the simulator. Click on the grey triangular button under 'Run' and watch you robot move over the field.
9		Use the blue button to return to the FlowCodeSheet.
10		When you click on the stopwatch icon, the properties of the stopwatch appear in the properties box on the right. Change the time in 'Delay' and experiment with different values in the simulator. The idea is to let the robot start at the bottom of the field and stop on the black line. You will have to count how long it takes to drive there. You can also add other Driver icons to make the robot move in different directions. Do not spend too much time on this. There is more interesting work waiting for you in the following assignments.

2. Finding the Black Line

Measure the black and white values



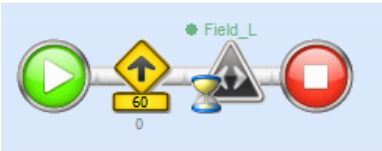
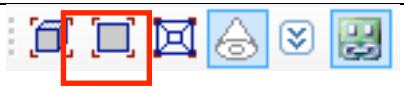
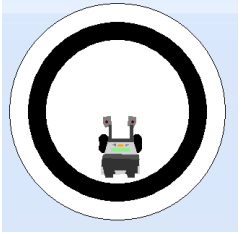
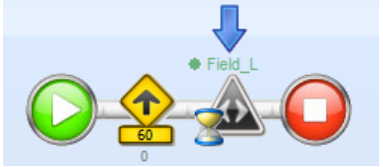
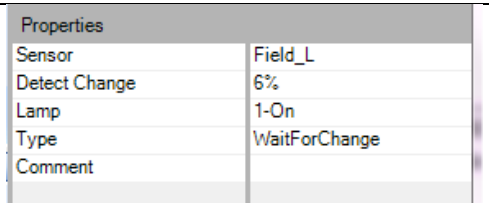
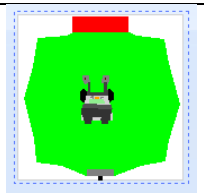
Just driving forward is not very interesting. We want the robot to use its ground sensors and react to what is sees there. So, while the robot is driving forward, it keeps looking at the ground and as soon as something changes, it needs to stop.



We can use this information to help the robot find out where it's going. This is much easier than having to count how long it takes to drive somewhere. Moreover, using a counter is not very accurate because when the batteries start running low, the robot will slow down. Also, if the robot starts out from a different place, things will often not work out as expected.

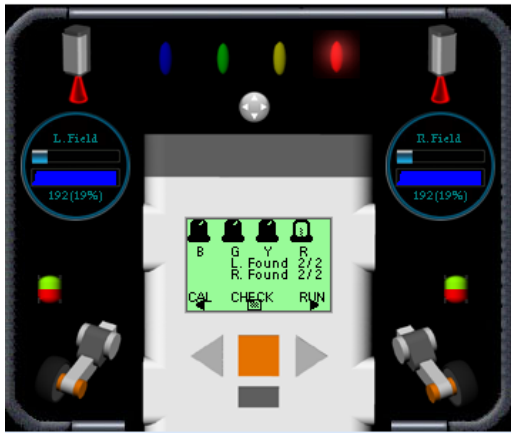
If we have the robot react to what it sees with its sensors, we can make better programs and low batteries or a different starting position will no longer have any influence on its operation.

Change your program: make the robot use its sensor to detect the black line. It will now react to every colour change on the field. It can find the black line, but it can also react to other colours.

1		Replace the Stopwatch in your program with a WaitForChange icon. You can delete the stopwatch by selecting it and pressing the Delete button on your keyboard. You can also use the Edit Menu and select Delete there. You will find the WaitForChange icon in the Sensor Flow Toolbox. While driving forward, the robot will check to see if the colour of the ground has changed. When you click the WaitForChange icon, its Properties will appear on the right. This indicates how large the change in colour must be before the robot reacts. You can use this to detect other colours.
2		Start the simulator with the green 'Run' button (do not forget to click the RobotWorldSheet first). Look at the row of icons right above the 'Control Panel' on the right. The second icon lets you view the field from above and makes it easier to see where you robot actually is. Also try the other two icons that change how you view the field.
3		Use your mouse to place the robot at the bottom of the field. As soon as the robot starts moving forward, it will start checking for the line.
4		Watch the blue arrow. It shows you what part of the program the robot is executing. As the robot drives over the white field, it is looking for a change in colour.
5		If you click on the WaitForChange icon, its Properties will appear on the right and you can tell the program how large the change needs to be. The 6% value makes your robot react very quickly. Make the value higher and check at what value it no longer sees the black line. You can experiment with other values, but you will need other colours on the field to do this. So go to the RobotWorld sheet and click on it to show the Toolbox for the World. Select EndTile from Rescue Premier: it contains colours grey, green and red.
6		You can now experiment with other colours as well, but you first need to find out what value the change percentage needs to be to detect the red area. Try to make the robot drive over the green field and stop on the red strip. You will need to move the robot over to the new field and remove the circle field if you want to test this.

3. Calibrating Colours

Measure the value of black and other colours

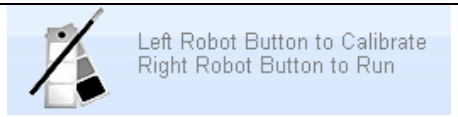
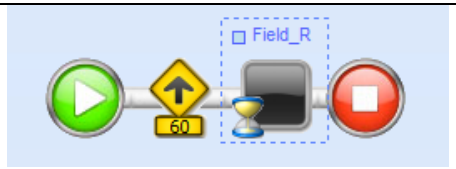
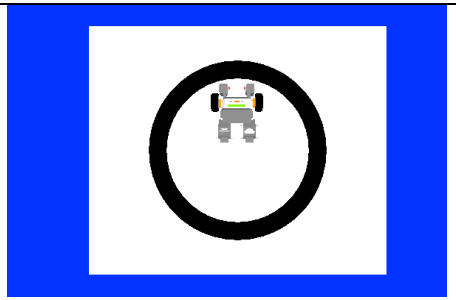
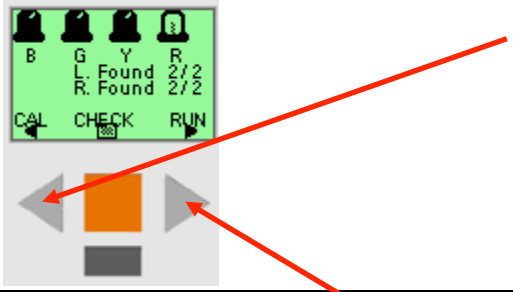


The robot can recognize colours, but you will need to tell it when to react to a colour every time. To do this you need to know about the change value of other colours. The sensors of every robot are all a little bit different, so you need to determine this value for every robot and every sensor. Not the best of situations.

It would be nice if the robot could find this out by itself. To do this, let the robot move forward and detect the different colours. The robot needs to remember the values of these colours. Measuring the colours on the field is called 'calibration' or adapting to the situation.

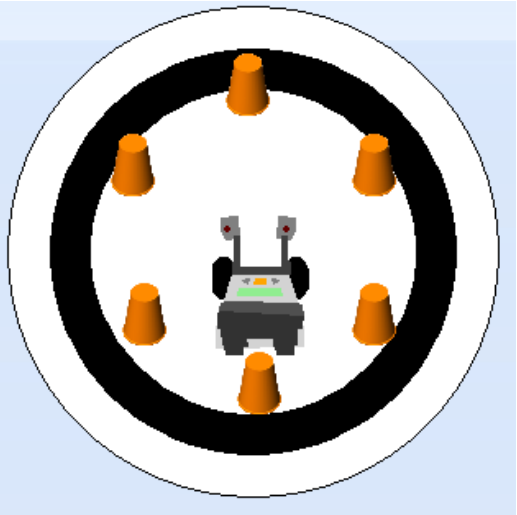
When the simulator starts, the CAL and RUN options will appear on the NXT screen. If you press the left button, the robot will start moving forward and memorising the values of the colours it meets. Please note that the calibration icon in the left top corner is needed to show the CAL and RUN menu.

If you press RUN, the program starts running as in the previous assignment. The program will use the colours it found during the calibration process. In this example, the robot no longer checks for a change, but only reacts if it finds the calibrated colour.

1		Click on the FlowCodeSheet and get the AutoCalibrateBW icon for black/white from SensorFlow. Place it above the icons of your program in the upper left-hand corner. This is a separate part of the program that is not connected to the other icons (also see page 3). If you forget this step, the CAL, CHECK and RUN menu will not be displayed.
2		Use the WaitForCalibratedBlack icon in your program instead of the WaitForChange one. This is the colour that will be detected as black during calibration. You can detect White in the same way. Use the right sensor by clicking on the icon and selecting its properties on the top line. You can choose either sensor by following the same procedure.
3		If you have used the EndTile icon with the green field, you will need to replace it with the circle field. Start the simulator and place the robot close to the black line. There are two sensors in front of the robot. Make sure that when the robot starts moving, it sees white first and then the black line. Start the robot with the grey RUN button. Now, you can select CAL, CHECK or RUN from the menu.
4		This is the CAL, CHECK and RUN menu. Press the grey button under CAL and the robot will start moving. It will stop automatically and display how many colours it has found for the Left and Right sensors. If you have placed the robot too far back, it will stop before it has found the black line. This means it will not identify two different colours. Make sure that both the left and right sensors have found two different colours. If the calibration procedure is not correct, the robot will not be able to stop on the black line. So if it does not identify two colours, you need to repeat the procedure.
5		Place your robot in the middle of the field and press the grey triangular button under RUN. The robot will now move forward and stop on the black line. The blue arrow shows you what part of the program the robot is executing.

4. A Small Game

Pushing the cups out of the circle

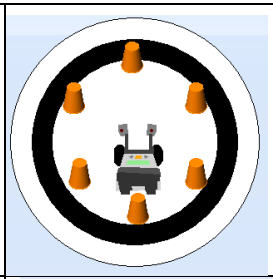
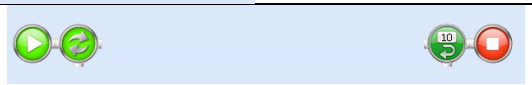
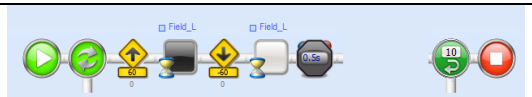
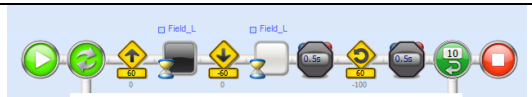
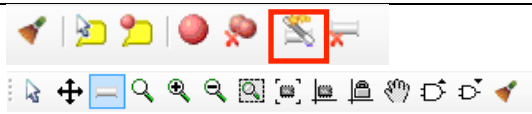
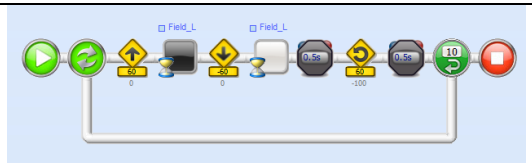


When your robot correctly stops on the black line, you can place a number of cups or cans on the edge of the field. Make the robot search the field and push all the cups out of the circle.

You can do this by having the robot move forward until it reaches the black line. Once it detects the black line, it must move back onto the white field and then look for the black line again. Repeat this a number of times by making the robot turn slightly each time, so that it can cover the entire field.

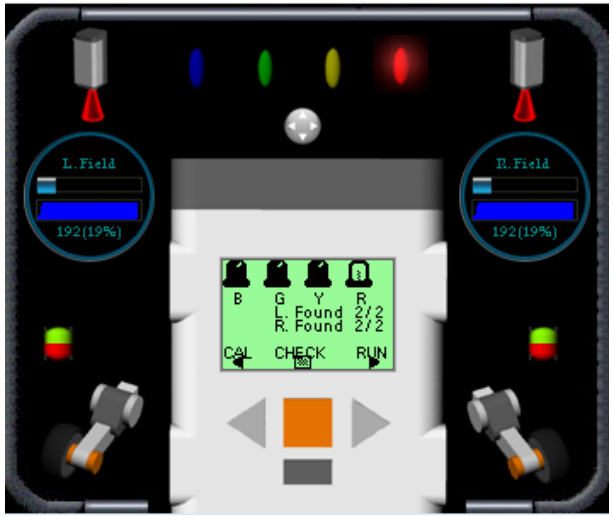
This is our first program Loop.

Test this program and, if it works fine, you can work on the last lesson in which you will load your program onto the robot's memory.

1		Make a new program with the circle field. Select the cups or cans from 'Basics and Rescue' and place them on the field. The robot has to push them out of the circle.
2		First, add a 'Merge' icon to the Program Flow and place a 'LoopCounter' icon at the end of it. You will develop your programme between these two icons. We want to repeat this part 10 times. This repetition is called a 'Loop'. Do not forget to include the 'AutoCalibrateBW' icon, so you can calibrate the robot (see the Sensor Flow).
3		Make the robot move forward until it sees the black line: use the WaitForCalibratedBlack icon to do this.
4		When the robot reaches the black line, make it move backwards until it is back on the white field and then make it back up a little more.
5		Make the robot turn a little, so that it will move forward in a different direction. The Loop repeats the program 10 times, allowing the robot to crisscross over the field and push all the cans or cups outside the circle.
6		Now, you must connect the lower ends of the Merge and LoopCounter together to complete the loop. There is a special Pipe cursor, which draws lines or pipes, to do this. However, as it is not so easy to use, we have made a Pipe Wizard that will fill in the missing lines auto-magically.
7		Use the Wizard to complete the loop between the Merge and the Loop icons and your program is ready. Try it out on the simulator. Change the value in the stopwatch icon to make the turns larger or smaller. This will change how your robot searches the field.

5. The NXT Robot

Loading the program onto the NXT



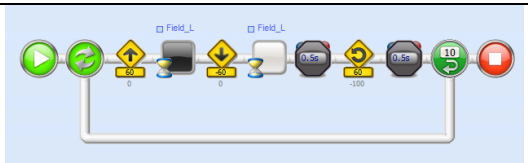
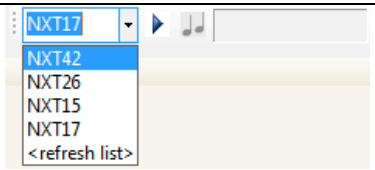
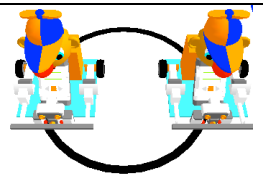
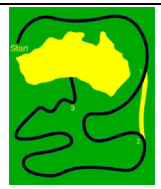
You have already seen the brain of the robot through the simulator control panel. We call this the NXT brick. It is the computer of the robot. We will now load the program onto the robot's memory and use the robot on a real circle field. To do this, you will have to connect your computer to the Lego NXT robot.

Load the program onto the robot by following the directions on the back of this sheet. Once you have loaded the program, you can use CAL and RUN to start the program on the NXT just as you did in the simulator.

Once a program is loaded onto the robot, you can start it by selecting RUN from the main menu on the NXT. This will start the last program that you loaded. If you want the robot to stop, press the orange button. If it does not react to this, you can switch it off by pressing the orange and the dark grey buttons under it at the same time.

You will notice that the NXT reacts slightly differently on a real field. It will have more trouble with different lighting conditions and shadows, as well as reacting to spots on the field. A certain amount of testing is required to make the robot work correctly. You can also add some strips of coloured paper to the field and see the robot react to different colours. You will have to calibrate the sensors for each new colour.

Try out the assignments in lessons 2 and 3 with different colours or thinner lines and see what happens with the real robot.

1		Make sure you use the same program. This program will be loaded onto the storage of the NXT robot. Start the simulator and check that your program still works correctly.
2	 	Put the RoboPAL dongle (usb stick) into a free USB slot on your computer. Then, start the Lego NXT robot by pushing the orange button. Watch the NXT screen: when it starts up you will see the number of your robot appear (i.e. NXT01). Remember this number, you will need it for the next step.
3		Look at the small boxes above the field in the simulator: the left one has a tiny arrow on the right side. When you click this button, you will see a list of all the NXT robots the computer knows. If your robot is not listed, use the bottom <refresh list> option. When you select this option, the computer finds out which NXT units are turned on. Select your robot from the list (i.e. NXT42 in the example).
4		Push the little black triangle next to the selection box to send the program to the selected NXT. In the box on the right, you will see an orange progress bar indicating that your program is being uploaded to the robot. When the transfer is completed the NXT will emit a series of beeps. If this does not work, turn the NXT off, back on and try again. Sometimes, the program is still running and the computer cannot contact the NXT. Switch the NXT off by pushing the orange and the dark grey buttons underneath it at the same time. If it still does not work, take the dongle out of the computer and put it back in and try again.
5		First, calibrate the robot on the field; place it in the middle of the field and press RUN. Check that your robot works correctly. If not, use the FlowCodeSheet editor to change your program. Always switch your robot off when it's not in use, as the batteries will run out pretty quickly.
6	 	If the robot works correctly, you can make it move faster or try a narrower line and check to see when it makes a mistake. The robot should be able to detect the line long enough to identify it; otherwise it will run past it. You can also try out some other colours by placing strips of coloured paper on the field. If it all works correctly, you are ready for the next set of lessons: choose Dance or Rescue. In Dance, you will make a robot dance to the beat of music, while in Rescue, your robot has to follow a black line on a green field to reach the yellow swamp and remove a 'dangerous' container (not really dangerous!)

Introduction for Teachers

Basics

The RoboDidactics workshops are based on individual lesson cards. This allows each student to work at his or her own pace.

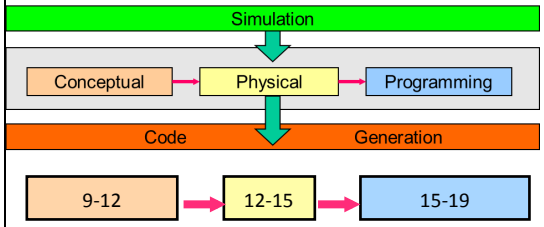
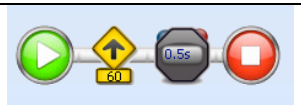
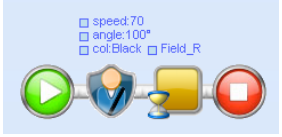
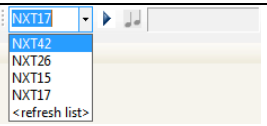
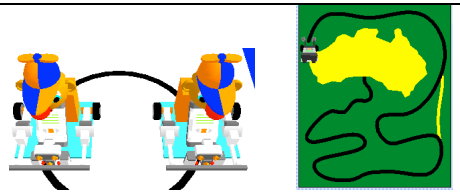
This is the Basics Level 1 workshop and is part of a series including:

1. Basics – basic concepts of RoboPAL programming and use of the Lego NXT robot
2. Dance – how to program a Dance robot
3. Rescue – how to program a Rescue robot

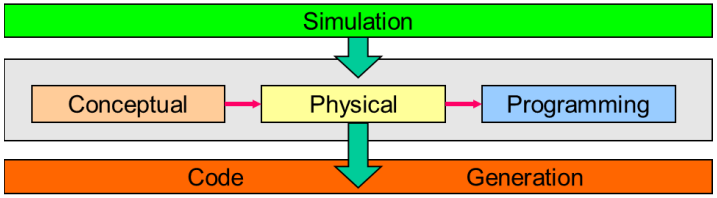
The picture on the right lists all the workshops included in the Conceptual Level (Level 1) together with their features and the individual lessons treated by each workshop.
After having completed the Basic lessons, students can choose the Dance or the Rescue Mission (or both). Many schools start with the Dance workshop and then move onto the Rescue workshop. The Basics workshop includes the following lessons, each treating an important idea:

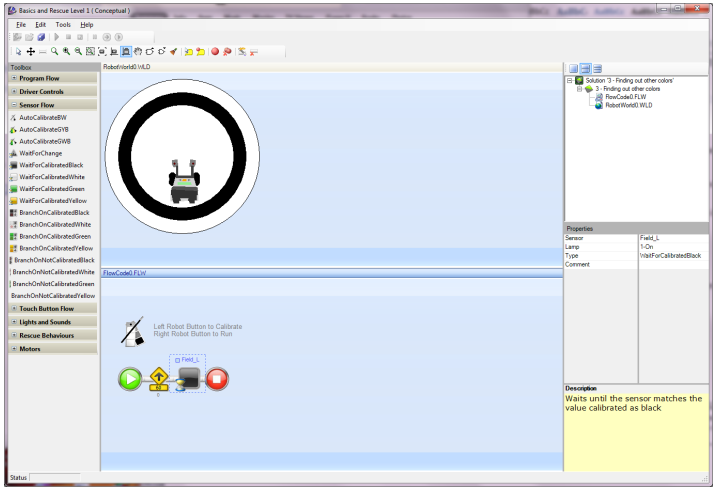
	Lesson	Concept	Features
1	Make your robot move forward	Movement	Making the robot move
2	Detecting a dark line	Calibration	Using a sensor to detect a line
3	Detecting other colours	Sensors	Using a sensor to detect colours
4	A small game	Loops	Driving across a coloured field
5	Loading the program	Uploading	Running a program on the NXT

Setup of the course material

1		Each sheet includes an explanation about what will be done during the lesson on the front side. A brief description minimizes the amount of reading that students need to do. The RoboPAL software is available for the Lego NXT robot.
2		An important part of the teaching material is the use of the integrated simulator. This allows students to see the result of their program on a PC without having to use a real robot. It makes testing easier and also reduces the number of robots needed in a classroom. In addition, students can develop and test their programs at home. The Conceptual Level concentrates on WHAT needs to be done. The Physical Level concentrates more on HOW things are done, while the Programming level concentrates on how to CODE a program in Java.
3		Programs are created using easy to understand icons in Levels 1 and 2; in Level 3, programs are coded in Java. Each Level adds new capabilities, while Dance, Rescue and Soccer have their own lessons and facilities as well.
4		Programs are simulated on the PC and can also be run on a NXT robot. Once a program is ready and running, it can be uploaded onto a NXT using a special dongle (usb stick) that takes care of all complicated software installation and driver issues.
5		For Rescue, a simple line-follower is provided to show students what a line-follower is and how they can influence its behaviour by changing the steering and speed at which a robot follows a line. Once students have mastered the rescue mission, they learn how to make their own line-follower with the Physical Level.
6		It explains how a program is loaded onto a robot and how the program runs on a Lego NXT robot.
7		After each lesson, students are stimulated to reflect on what they have learnt and what the program does. They do not merely follow instructions, but gain a true understanding of what they do. Once the students have completed this set of lessons, they are ready to participate in the Dance, Rescue or Soccer competitions of RoboCup Junior.

The basic course for Level 1





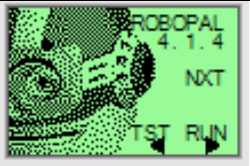
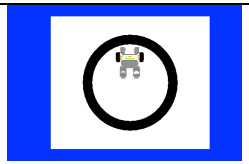
The instruction cards are part of the RoboDidactics learning track.

Students learn to use RoboPAL, the integrated simulator and the Lego NXT robot. These lessons assume that a robot has already been built and that the RoboPAL firmware has been uploaded.

Instructions on how to build the robot are available separately. In the teacher's guide, a description is given on how to install the software and how to load the RoboPAL firmware onto the NXT.

Information for teachers and installation instructions can be found at www.robocupjunior.nl in the download section and also at www.robopal4nxt.com

Before the lessons begin a number of things have to be prepared. Please read the back of this instruction card for the step-by-step instructions.

1	 RoboPAL4NXT.lnk	Load the RoboPAL program onto the students' computers. The application does not need to be installed: just copy the executable file from the CD or the download area of the website to your computer(s). Place a shortcut to the program on the desktop, so the program can easily be found.
2		Make sure that you have one robot for every 4-5 students. The building instructions for assembling the robot can be found on the RoboPAL CD. The students can do this themselves by following the instructions during the course of a lesson. The instructions can also be found at www.robocupjunior.nl in the download section.
3		Make sure the LeJos firmware is installed onto the robots that use RoboPAL. After installing the LeJos operating system, the RoboPAL firmware needs to be loaded. Installation is done through RoboPAL. The instructions can be found in the teacher's guide. Students should only do this part of the installation if they are using their privately owned Lego NXT robots. When installing firmware on a NXT, the Lego NXT driver must also be installed on the computer.
4		After installing LeJos on the NXT, the file RoboPAL.nxj must be selected as the default start file. Then, make sure that the NXT Bluetooth is on turned via the LeJos main menu. Finally, set the System to Auto Run, after which the robot will always start up with the RoboPAL menu. This prevents students from using the LeJos menu and erasing important system files from the NXT.
5	 	When using the Basics and Rescue lessons, make sure you have created a Circle field or a Rescue field. The Circle field can be made out of a large sheet of white paper. Make the black line using a felt-tip pen or black insulation tape. Make sure that the line is at least 2 cm wide. For the rescue lessons a special rescue field is required. Instructions for making it can be found on the RoboCup Junior or RoboDidactics sites.
6		The teacher should already know all the lessons and, even better, the entire Robotics module for NXT. This material consists of three parts. The first part handles the basic course for the Physical Level and includes Dance and Rescue. The second part treats the Rescue mission. The third part is only necessary if the students are also going to move on to the Level 2 courses. This basic module is the precursor to the courses on Dance and Rescue.
7	 	In the last lesson of the basic course, students learn how to upload their program onto a NXT robot. To do this a special RoboPAL dongle is required. You will need at least one, but to avoid having all the students sharing the same dongle, one per robot is advisable. Before using the NXT in the classroom, be sure that the PC will connect to each robot. Once the students have learned how to upload a program, they can test all their programs on the NXT robot.