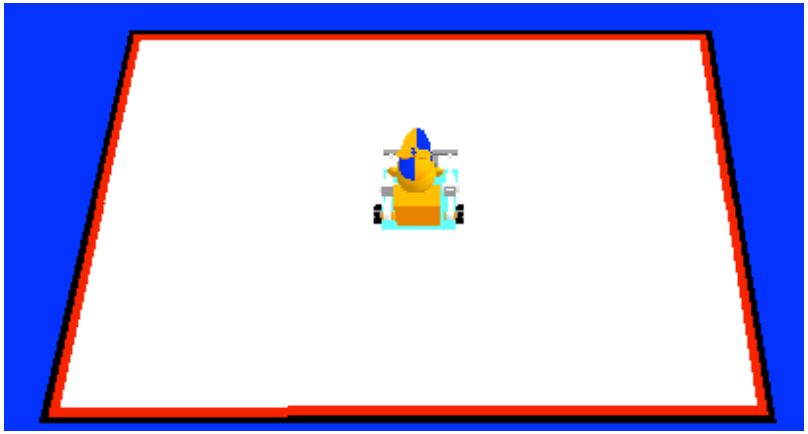
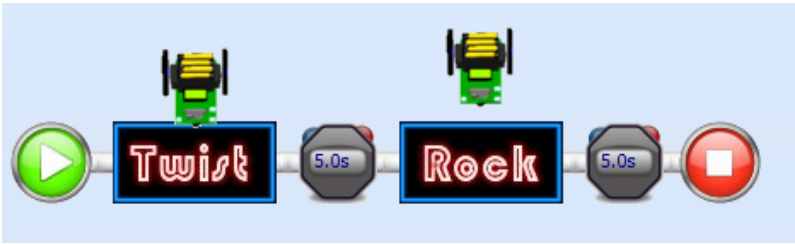
1. Let's Dance

A simple dance routine introduces two built-in dance patterns

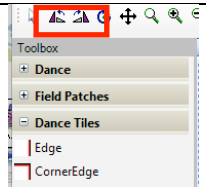
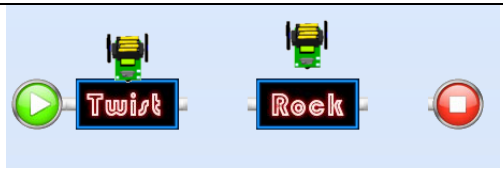
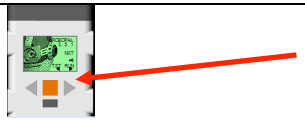
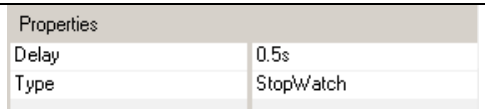
Making a robot dance to the rhythm of music is a very nice way to learn about robotics. Dance movements are an excellent way to learn how to repeat certain patterns of movement. We have already included some simple patterns (Twist and Rock) and this course will teach you how to make your own patterns.



With Dance, patterns are what your program is all about, so we will show you how to make and use patterns.
The built-in patterns are easy to use, but naturally it is important that the patterns are performed at the right time, so that it looks like the robot is really dancing to the music.
As the robot does not hear the music, you will have to program the timing. Your basic job is to select a piece of music and design patterns that suit it.

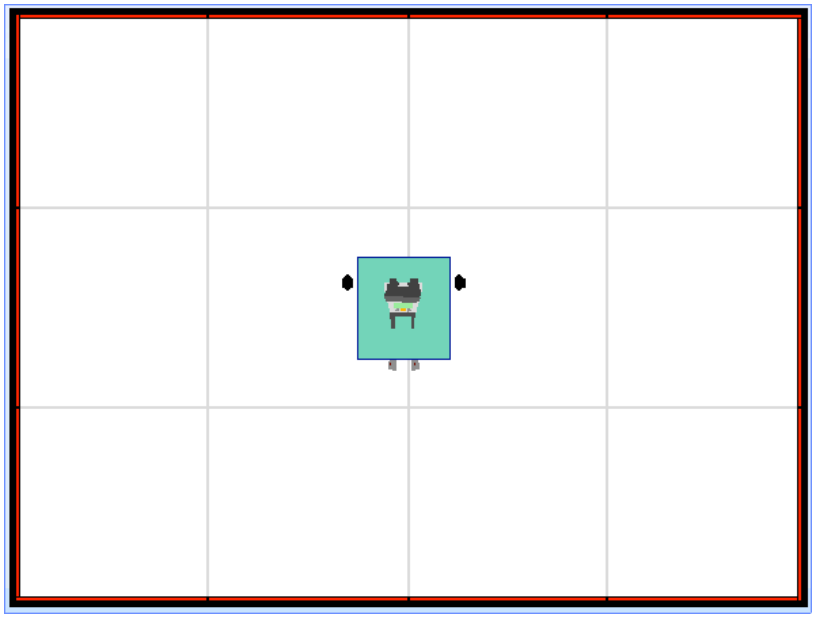


Read the instructions on the back of this card and make your first dancing robot program.

1	 RoboPAL4NXT.lnk	If you have not started the program, do so now by double clicking on the RoboPAL icon on your desktop. If the program is already running, go to step 2.
2		If you do not see this screen, choose “File | New” from the File menu. Then, select ‘Quickstart new Dance Field Project’ and open it. Click on the bottom window with the green and red icons. This is the FlowCodeSheet in which you will be making your program. If your computer has a small screen, part of your program may disappear on the right-hand side. Use the hand cursor to move the screen or the zoom cursor to adjust your program size.
3		Enlarge your Dance Field by the Dance Tiles from the ToolBox in the RobotWorldSheet into the world. You can use the rotation icons to rotate the tiles. You can make the dance field as big as you wish. The official Dance field is 6x4 meters or 20x13 tiles, but this is far too large a size to use on the simulator. So, in most cases, we limit the field size to 6x4 tiles. Remember, however, that in the real Dance performance your field will be the official size (6x4 meters).
4		Click on the FlowCodeSheet and select the Twist and Rock Dance Moves from the Toolbox. Place them after the start icon, leaving some space between them to add other icons.
5		Select a Stopwatch icon from Program Flow and put one after each Dance Move. This tells the program how long the dance move lasts. As we have no music at this point, it doesn’t really matter how long the moves last.
6		Go to the top of the RoboWorldSheet window and click on the green ‘Run’ button to start the simulator. You will see the dance field and, on the right, a picture of the NXT.
7		Press on the grey triangular button (under RUN in the screen) to start the simulated NXT robot. Watch your robot move over the field. You can also move the robot over the field by using the Mouse and you can change its direction using the right mouse button.
8		You can restart the program by using the blue button with the arrow or you can exit the simulator with the small blue button.
9		Back in the program, click on the stopwatch and check the ‘Properties’ window on the right. Change the time in the ‘Delay’ field and experiment with it on the simulator. Make your robot perform each Dance Move for different amounts of time. This is your first Dance program.

2. Dance Moves and Your Robot

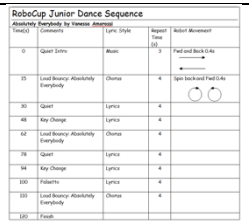
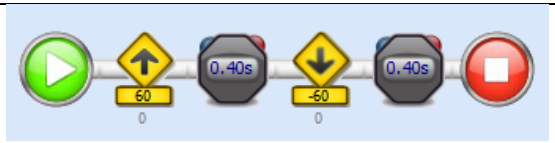
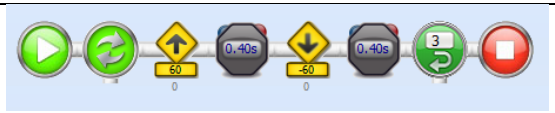
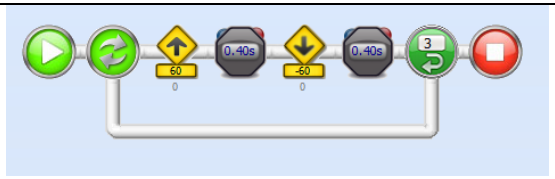
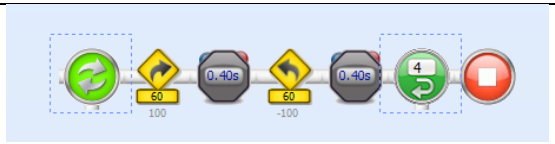
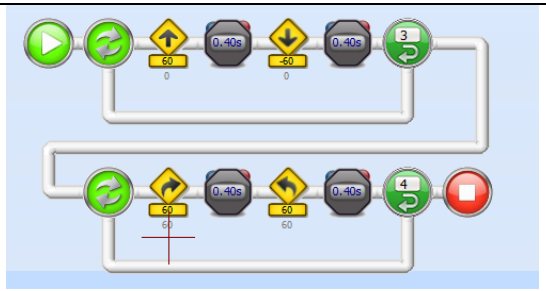
Things to watch for when making your robot dance



You will start practicing with your robot on the large Dance field. The programming task for a dance robot is to make it perform interesting moves. There are several ways to do this:

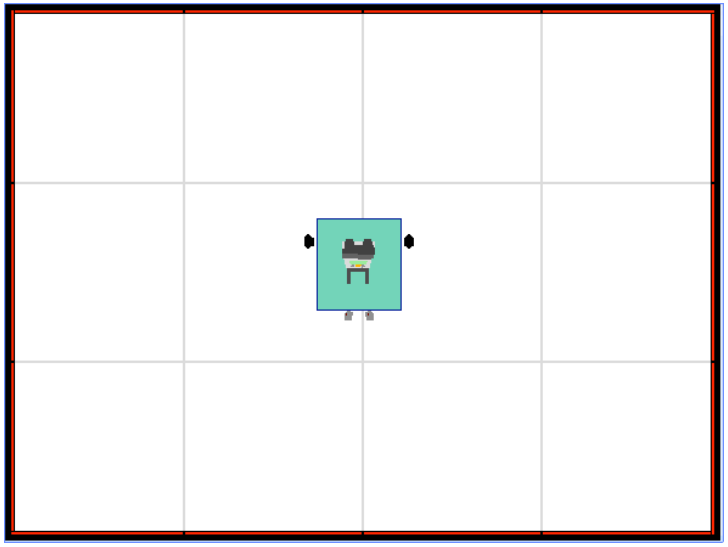
1. Use the built-in Dance Moves.
2. Create your own Dance Moves, using motor drive commands and timing diagrams. This is the first thing we will show you how to do.
3. Draw a pattern on the floor and make your robot follows it.
4. Use a MIDI file. (These two last ways are more difficult and will be addressed in the other lessons.)

Let's see how to make your own Dance Moves. As the Lego robots are rather small, you will have to create extensions to make a bigger and more stable robot. There are several ways of doing this. We will show you in a separate set of instructions. The robots you see on the simulator use a base plate connected to the motors and sensors. This gives you a stable base on which to build a larger dance robot that will not fall over so easily. First, you need to select your music and design your choreography (the dance moves that your robot will perform). As part of this lesson, you will receive a special form on which you can draw your dance moves.

1		Take the Dance Sequence sheet that is part of this course and select a piece of music for your robot. Find out where the chorus and the lyrics are in your music and write the timing on the form. Now, you are going to make a choreography in which you decide what moves the robot must perform during each part of the music. You can decide to use the built-in Dance Moves like Twist, Rock, Boogie or Woogie. This is the easiest thing to do, but it is far more interesting to make your own Dance Moves. We will show you how to do this.
2		Make a new Dance program and call it 2 – Dance Moves. We will follow the example here, but the idea is to create your own dance move and program it. The first step makes the robot move forward and backward for 0.4 seconds. This needs to be repeated three times. So, create the forward and backward movement. You should already know how to do this.
3		This movement needs to be repeated three times, so we will use a Loop. Select a LoopCounter from program Flow and place a Merge Icon at the beginning of the program. Set the LoopCounter to a value of 3 via its properties.
4		Connect the Merge Icon with the LoopCounter by using the Wiring Wizard, the last icon in the top Menu Toolbar. You have already seen how to do this in the Basic lessons. The Wiring Wizard will connect the two icons auto-magically.
5		The second step is to repeat rotating left and right movements for 0.4 seconds four times. So create a new group of icons (like the previous one) underneath as a separate Dance Move. Later on, you can decide how sharp the turns should be, but for now use the standard SwerveRight and SwerveLeft icons. Move the Finish Icon from the Dance Move above to the one below.
6		We will use the Wiring Wizard here again to connect the two parts. First select the Merge icon on the first row and then, select the LoopCounter Icon while keeping the Shift button down. (as seen in step 5). That will select the two icons. Now use the Wiring Wizard to connect the two selected icons. You then need to connect the first Dance Move to the second one. Select the LoopCounter Icon in the first row and the Merge icon in the second row and use the Wiring Wizard again. (There is a separate video that explains more about wiring). Use your Dance Pattern worksheet to develop the remaining Dance Moves. Then, save your program and test it on the simulator. If it works correctly, make sure it also works with the robot. Make sure that your robot always moves back to the centre of the field, so that it does not go over the field border. You may want to change the Steering in the Swerve Icons to make it Spin left and right in a circle.

3. Staying Inside the Field

Making sure your robot stays inside the field

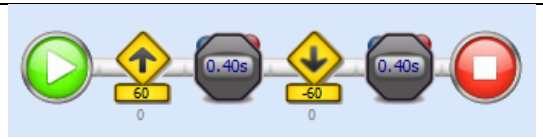
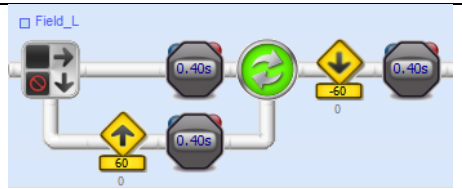
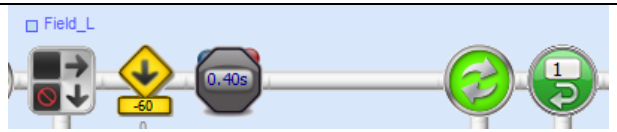
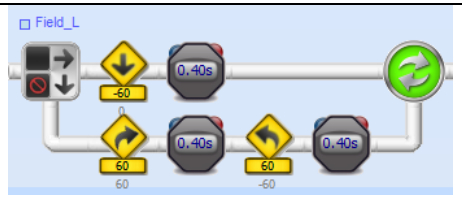
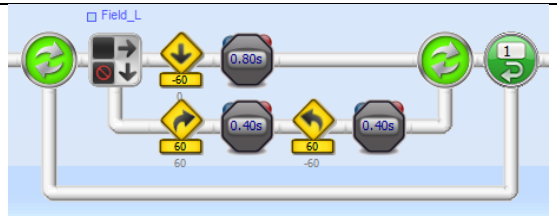


A black and red line surrounds the Dance field. You must make sure your robot does not move outside the field. The simplest way to do this is to start in the middle and make the robot move forward and then backward just about as much, but a better way is to make the robot use a light sensor to detect the black line.

This is not so simple, however, because you must make sure that the robot only moves forward if it does not detect the black line.

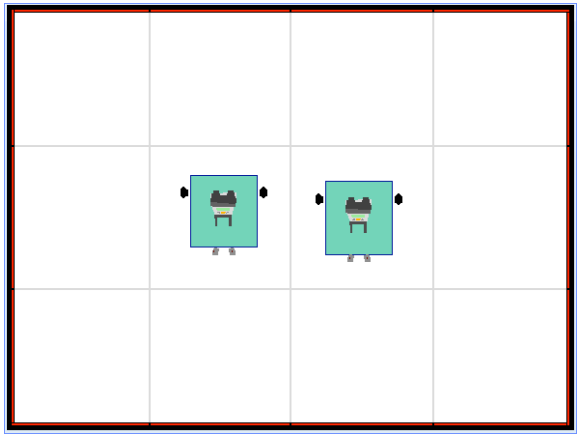
You will have to check that your robot does not drive over the black line for each Dance Move. In the Basic lessons, you learned about calibration and detecting the black line. Now, you will use these skills here.

We will modify the Dance Moves to always check if the move can be executed first. If not, the robot will move backward to stay on the field. However, if an expected move is skipped, this will change the Dance patterns. So make sure that your robot stays inside the field during testing. If it detects the black line, change the Dance Moves in your program. Please note that the robot can only detect the black line while moving forward. It cannot detect the line when it is moving backward.

1		When the robot performs the first move, it will move forward and then backward. We must modify this behaviour to check if the robot detects the black line and, if it does, skip the forward movement.
2		To make sure that the robot stays on the field, it must always check to see if it can complete a pattern by moving forward. If the sensor detects the black line, it skips the forward motion and tries again. We use the BranchOnCalibratedBlack icon to do this, but this creates a problem as it changes the timing. So you need to include a stopwatch, in the part where the robot does nothing, so it pauses as you can see in step 3.
3		When the black line is NOT detected, the forward motion can be completed. In BOTH cases, the robot can safely move backward. However, if robot is at the far end of the field, it could drive backward over the black line. As it has no sensors there, it would not be able to detect this. So our programme makes sure that when the black line is detected, the robot will only move backward (rather than forward and backward as before).
4		In the second move, we cannot do the same thing as in the first. In the first move, we allowed the robot to move backward when it detected the black line. If the next move does not include a backward movement, but a sideways movement, we must include a backward move to get the robot away from the line.
5		The robot will move backwards only if it detects the black line. The robot's normal behaviour - when the black line is NOT detected - should be executed in the lower NOT exit of the BranchOnCalibratedBlack icon.
6		This is the complete wiggly move. When the black line is NOT detected, the normal move is executed; only when the black line is in sight, does the robot move backward instead of performing the wiggly move. This, however, does influence the dancing behaviour, as the move only lasts 0.4 seconds in a single loop. This means we must change the 0.4 seconds to 0.8 seconds to make the dance move fit the same timing.
7		The second dance move looks like this. Notice that when you use the selection box from the StopWatch icon, it does not list 0.80 seconds. You can type in any value you like. Now try this program on the simulator and, if it works correctly, test it on a small dance field and make sure it does not cross the black lines.

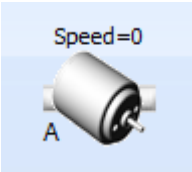
4. Moving the Head

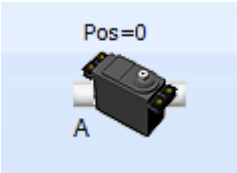
Adding extra movements



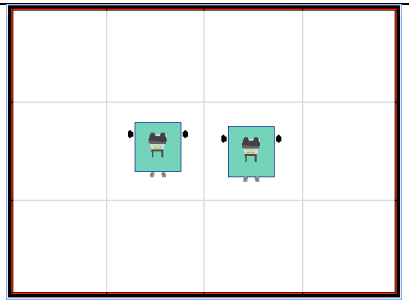
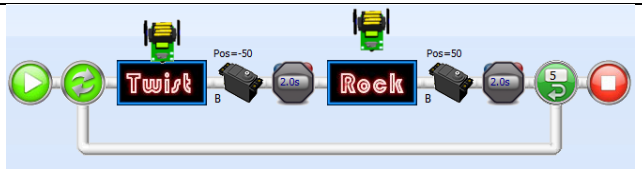
So far, our dance movements have only included forward and sideways movements, but it would be nice if the robot could also move its other parts. In its standard form, the NXT only supports three motors, so we will now concentrate on the third motor. It is possible to add extra motors and lamps to your robot, but they cannot be controlled via the program and need to be powered externally.

The motors of the NXT can be used in two different modes: Standard or Servo. The Standard mode is what we have been using so far: you determine the speed and direction of movement of this type of motor. A Servomotor, instead, works in a different way. It allows you to define the position that the motor needs to take. If you are using an arm or a head, for instance, you may like it to be positioned to the right, to the left or in the middle. A Servomotor allows you to specify the positions for left, middle and right by using percentages: -100/0/+100. You do not have to count how long the motor is running. It will set the motor to the desired position immediately. You can do this with the ServoMotor icon.



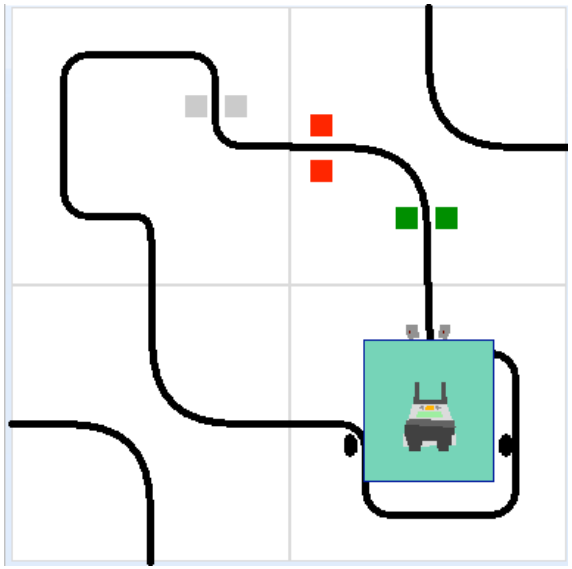


We will show you, in this lesson, how to use the third motor as a ServoMotor and control the position of the robot's head. You can also change the type of motor and make the head spin by using the standard (DC) motor instead. To make things even more interesting, let's make two robots dance at the same time with the same program.

1		Place a second robot on the field by selecting another LegoNXT robot from the ToolBox in the World. They will both execute the same program. The robots can also do different things, but this is addressed in Level 2.
2		First make a program with a simple dance pattern using the built-in moves. Make the program repeat 10 times. Test it and see how the two robots dance to the music. Notice that we have included a new part called DanceMusic. This contains a MIDI file that actually plays the music. We will see more about MIDI in lesson 6.
3		Add a servomotor from the Motors toolbox to the program and set the positions to -50 and 50. This will make the robot's head move from left to right. You will need to specify where this motor is connected. As the two driving motors already use ports A and C, connect it to port B in the middle. Run the program and watch the robots move their heads from side to side. Experiment with the settings.
4		It is really a pity that the NXT only has three motors. Other motors can be added, but this requires extra hardware. However, there is another possibility. You can connect motors and lamps directly to an external battery. Although this adds more movements, your program cannot control them. Nonetheless, it will allow you to have a head that can spin or make other movements. You can also construct an arm and connect it to the third motor.
5		There are other possibilities, too. If you want your robot to just drive forward and backward, all you need is one motor. This leaves you with two motors to use for other movements. Make sure your robot does not drive off the field. Try your program with the real robot and see how it performs. There also exists a special extension set for RoboPAL that lets you connect additional motors.

5. Dance Tracks

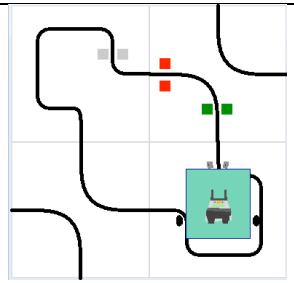
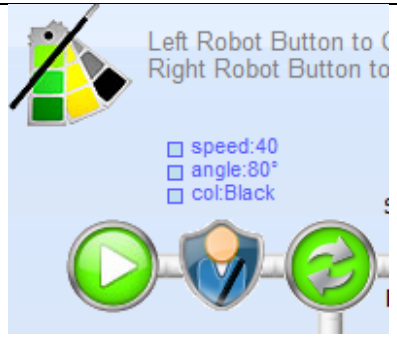
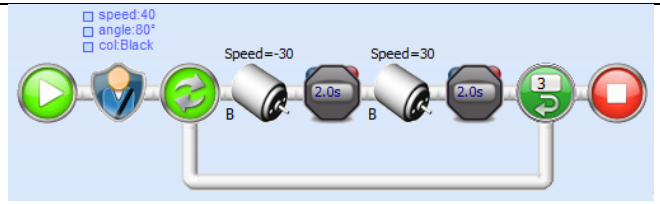
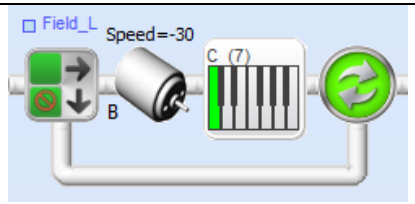
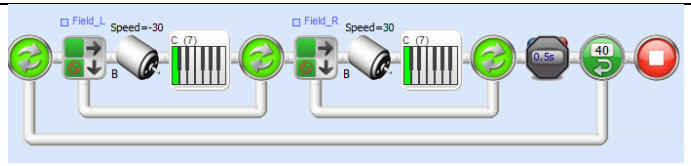
Making your robot follow a line



We have made a program that makes a robot dance by counting the beats in the music, but if we have several robots that have to act together, it may be much simpler to draw lines on a sheet of paper and place them on the dance floor. The robot can follow the line and change directions following the painted line.

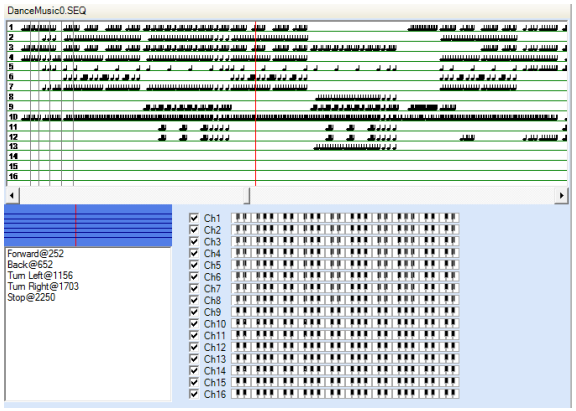
We are going to use a simple line-follower to do this. First, calibrate the robot so it can detect the black line; then, make it follow the line and perform the movements in your pattern. Just make sure that your robot is not moving too fast, so that it can follow the curves. If the robot loses track of the line, the entire performance will be ruined.

There are other possibilities, too. You can put some coloured spots on the field. When the sensors detect this colour, the robot can react using the third motor. The colours Green and Yellow can be used as signal functions: by calibrating for these additional colours, you can have the robot move its head or arm when it detects one of them. The nice thing about this is not only that you can test your program on the simulator, but also that by making your own pattern field, you can make the robot do anything you want without having to modify the program.

1		Make a new program: 5 – Dance Tracks. You can change the pattern in the example by replacing it with other tiles from the Dance Tiles Toolbox. You can draw any kind of pattern on a piece of paper. Make sure the corners are not too sharp. Note that when you are testing the programme on the simulator, sharper corners may cause the robot to lose track of the line. Make the black line at least 2 cm wide and make it dark enough so that it can be easily detected. Replace any of the tiles to change patterns and include Field Patches to make your robot react to different colours (see steps 4 and 5).
2		Select AutoCalibrate GYB from Sensor Flow. This will allow you to use all three colours. Then, select the simple line-follower from RescueBehaviors and place it right after the Start icon. The line-follower has three properties: Speed, Angle and Colour. As we are only using a black line here, we only need the Black line-follower. There are other options, but these are used in the Rescue mission. The speed tells the robot how fast it needs to move. For now, keep the speed at a fixed value. The Angle tells the robot how sharp to make its turns. The sharper the turns, the larger the angle should be: 100 is the maximum angle, like the maximum turn of a car's steering wheel.
3		You can also use icons to move the head, so that the robot looks around while it is moving. Place a LoopCounter at the end, so you can program how many times the movement must be performed. Moreover, it will also allow you to change the speed or other parts of the robot behaviour. If you want the robot to stop following the line, you will have to include a NoBehavior icon. Test this program on the simulator. Make sure you first calibrate your robot for the colour black.
4		If you want the robot to react to the colours it finds on the floor, put the colour on the field and then include a test in your program to see if that colour is present. Now, instead acting for a fixed amount of time, the robot reacts to the colours. You can use the left or the right sensor, which gives you a total of four possibilities. Use the BranchOnGreen or Yellow to test the colour.
5		In this case, when the left sensor detects green, it moves its head to the left and when the right sensor detects green, it moves its head to the right. You can use all types of patterns and movements in this way. Just make sure the colour is close enough to the line, so the robot can detect it. Also, make sure you include a stopwatch at the end to time your dance move.

6. Using Midi

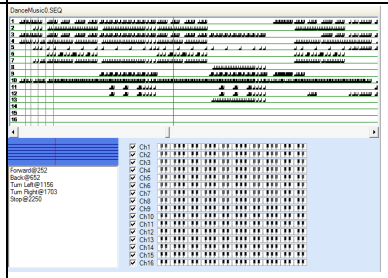
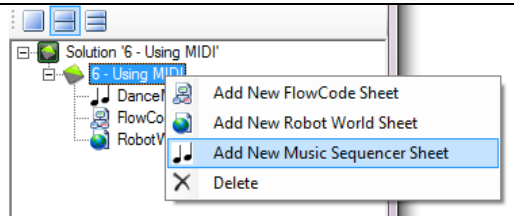
Introducing Midi events and coupling them to robot movements



If you want to play a musical instrument using your computer, you can use its MIDI facilities. MIDI is a technique that makes the computer play music by controlling a keyboard and pressing the keys to make music. The picture on the left shows the MIDI control panel of RoboPAL. You can download many different songs in MIDI format and include them in your program, by adding a new Music Sequencer Sheet. Use the sequencer to select a precise point and a precise instrument, for instance when the drummer hits the base drum.
When you play the music, a red line indicates what part of the music is being played. If you click on that point and select an instrument, RoboPAL creates a so-called event. You can connect this event to a movement performed by your robot and control its actions directly through the music. First, however, you need to:

1. Select the music and find a MIDI file to play.
2. Select the number of points at which you want a certain action to take place.
3. Click on these points in the sequencer to create a MIDI event.
4. Develop your program and connect the robot actions to these events.

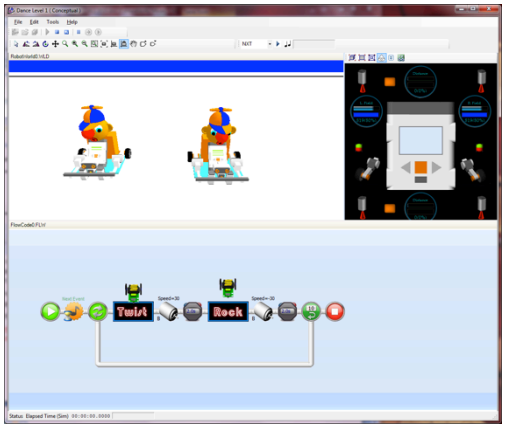
We will show you how to do all of this. Using a MIDI is the most difficult way to make a Dance program, but it also provides the most possibilities. So, if you really want to make a good impression with your dance performance, try out the MIDI and you will be surprised by how much you can do with it.

1		First, we must explain MIDI. This is a very difficult element, so we have made a special instruction video you can watch on your computer. It includes a PowerPoint presentation and an instruction video. When you have followed both, you should understand how MIDI works and be able to start making your program. So do that first and then return here.
2		Open program 6 – Using MIDI. Select the MIDI sheet and drag it to your worksheet. The Sequencer includes a ToolBar and a number of events that have already been defined. Run the sequencer, listen to the music and watch the events. On the bottom, you can see all the instruments that are used in the music. You can turn these instruments on and off to hear exactly what each instrument is doing. This makes it easier to select an instrument and the precise moment at which you would like your robot to do something. Of course, to make your own program you will have to select your own music and then create the MIDI events that you wish to use.
3		Drag the flowcode to the worksheet. All the events that have been defined are now in the sequence. Note that events must take place in the same sequence in which they occur in the music, otherwise you might be waiting for something that has already happened. An action is performed at each defined MIDI event.
4		This is actually a much simpler way to make the robot do what you want than using dance moves. It just takes a bit of reading to understand how MIDI works.
5		Run the program on the simulator. For your robot to react to the actions, the NXT also needs to have the MIDI file stored in its memory. So, in addition to the program, you also have to upload the MIDI file onto the NXT. Use the special button with the musical notes in the upload toolbar to send the MIDI file to the NXT and test the program on the robot to see if it works correctly.
6		If you want to make your own program and use different music, you first need to download a MIDI file. Then, go to the Project Browser on the right-hand side and click the right mouse button to show the menu. Select 'Add new Music Sequencer Sheet' from the pop-up menu and import the MIDI file you have downloaded there. You can then use your own music.

A Dance Performance

Putting it all together

So, now that you know how to program your robot, it's time to put your performance together.

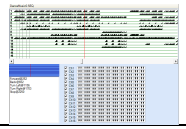


The Dance challenge consists of a number of steps, detailed on the back of this card. In general, the following things need to be done:

1. Select the music and design a choreography
2. Design robots, costumes and props
3. Build the robots and props and make costumes
4. Program the robots
5. Rehearse and keep track of what you have done in a logbook

The most important thing is to know about the rules to put together a good performance. You can concentrate on the theatrical content to tell a story (Type T) or you may concentrate on the performance in which the entertainment value is most important (Type D). Select one of the following options:

1. Music as part of the dance (D) or just as background (T)
2. Deliberate and accurate movement to music (D)
3. The theme and the story are the main issues and music just supports it (T)
4. The students take part in the show (T) or just in the background complementing the dance (D).
6. The scenery is more adequate to theatre (T)
7. More props (T)
8. Focus on choreography (D)
9. Continuity of the dance (D)

1	Select your music	Select your music. If you decide to use the hi-tech solution, find a MIDI file; otherwise, use a dance sequence sheet.
2	Select Dance, or	**'Dance'** is a performance closely sequenced to music: robots are required to move in time to the beat or rhythm of the music, in the same way that a human may listen to the beat of the music and dance to it. Dance assessment is closely focused on the choreography and synchronized movement of the robot(s) to the music.
3	Select Theatre	**'Theatre'** is used for a performance in which music is used as part of the performance, but robots are not required to move strictly in time to the rhythm or beat. 'Theatre' tells a story or develops a theme. Theatre assessment is focused on the overall theatrical presentation: how effectively the robot(s) present a theatrical item. Examples: nursery rhymes, Star Wars or movie-inspired presentations, Olympic games, cars, etc.
4	Design your robot 	Make a design for your robot(s). If you have more than one robot, your performance will be much more interesting. Also, make the robots much larger than standard Lego allows you. You may have to create a stable base and build a frame to support a larger robot.
5	Select how you will program your robot 	Decide which method you want to use to program your robot. a) The first time you may want to just use the Dance Moves. b) You could create a line-follower field with a line and coloured dots to make your robot perform. c) Study the MIDI method and decide if that will make your robot performance more interesting.
6	Build your robot 	Build your robot and design costumes for the performers that match the robot's design. You may want to look at modern techniques like laser cutters and 3D printers. Some countries have FabLabs where you can make your robot design using advanced production techniques that are available to schools.
7	Program your robot 	Design your choreography and transform it into your Dance Moves using the Dance Sequence Form. Then, program your robot and test it on the simulator to check the timing of the moves.
8	Test your robot(s) 	Test your program with the robot, using live music.
9	Check the Dance rules	Check the rules for the Dance competition at www.robocupjunior.nl or any other RoboCup site and find out how the performance will be judged.
10		Have fun building and testing your robot. Make it interesting for the robot builders, the choreographers and your audience.

Introduction for Teachers

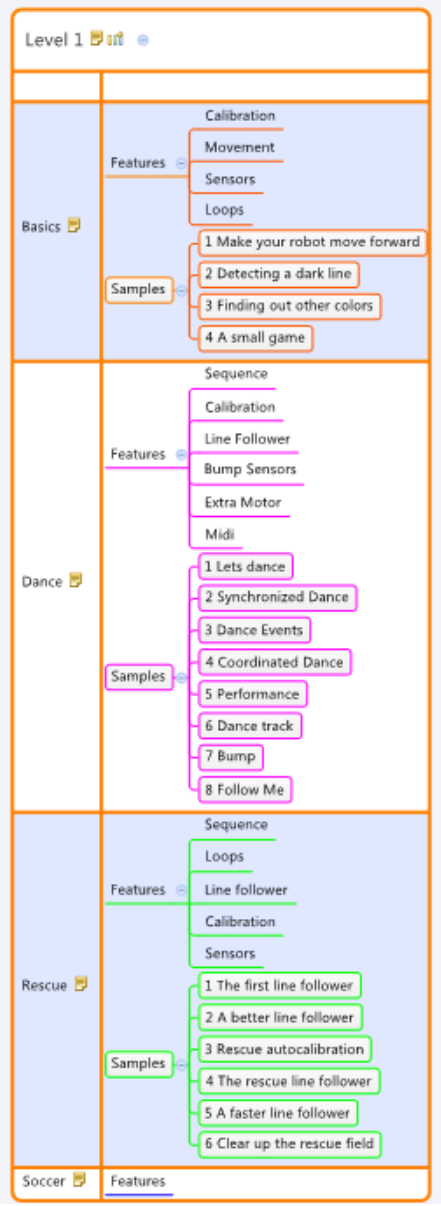
Dance

The RoboDidactics workshops are based on individual lesson sheets. 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:

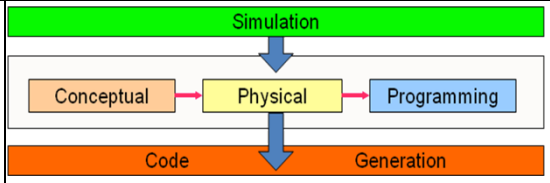
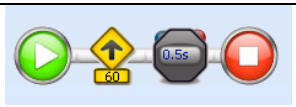
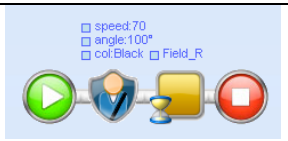
1. Basics – basic concepts of RoboPAL programming and use of the Lego NXT robot
2. Dance – how to program a Dance robot
3. Rescue – how to program a Rescue robot

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.
The Dance workshop contains the following lessons, each treating an important idea:



1	Let's dance	Simple dance routine, introducing two built-in dance patterns
2	Dance moves	Using built-in Dance Moves
3	Detect the field borders	Make sure your robot stays inside the field
4	Moving the head	Adding extra movements and using the third motor
5	Dance tracks	Two robots following a line pattern
6	Using MIDI	Using MIDI to define when a robot does something
7	A Dance performance	Putting it all together

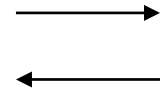
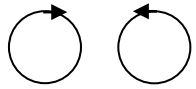
Setup of the course material

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 the students need to do. The RoboPAL software is available for the Lego NXT robot and for a special robot developed for these courses: the JoBot Nano.
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 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, while 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 the program is ready and working, it is uploaded to 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 the robot follows the line. Once students have mastered the rescue mission, they learn how to make their own line-follower with the Physical Level.
6		The students gradually learn about loops, subroutines and passing parameters. They also learn about calibration, testing and issues that have to do with timing and noise in sensors.
7		After each lesson, students are stimulated to reflect on what they have learned and what the program does. They do not merely follow instructions, but gain an understanding of what they have actually done. Once the students have completed a set of lessons, they are ready to participate in the Dance, Rescue or Soccer competitions of RoboCup Junior.

A Dance sequence is developed by using a form on which you can write down all the movements that your robot will perform. You can use the built-in Dance Moves or you can make your own. The total duration should be about 120 seconds.

RoboCup Junior Dance Sequence

Absolutely Everybody by Vanessa Amorossi

Time(s)	Comments	Lyric Style	Repeat Time (s)	Robot Movement
0	Quiet Intro	Music	3	Fwd and Back 0.4s 
15	Loud Bouncy: Absolutely Everybody	Chorus	4	Spin back and Fwd 0.4s 
30	Quiet	Lyrics	4	
48	Key Change	Lyrics	4	
62	Loud Bouncy: Absolutely Everybody	Chorus	4	
78	Quiet	Lyrics	4	
94	Key Change	Lyrics	4	
100	Falsetto	Lyrics	4	
110	Loud Bouncy: Absolutely Everybody	Chorus	4	
120	Finish			

RoboCup Junior Dance Sequence

Song Name:

Time(s)	Comments	Lyric Style	Repeat Time (s)	Robot Movement
120	Finish			