



ROBOESL PROJECT

ROBOTICS-BASED LEARNING INTERVENTIONS FOR PREVENTING SCHOOL FAILURE AND EARLY SCHOOL LEAVING

Erasmus+ 2015-1-IT02-KA201-015141

Output 1: Curricula for 10 exemplary interdisciplinary robotics projects

Curriculum 2: Go to park

Lead Partner: UNIPD (IT)

Authors: Michele Moro, Francesca Agatolio, Emanuele Menegatti (UNIPD)

Contributions

Dimitris Alimisis (EDUMOTIVA¹) (Chapters 2 and 3)

Linda Daniela (University of Latvia) (Chapter 5)

Circulation: Public

Version: Final

Stage:

Date: March 31, 2017

¹ EDUMOTIVA stands for 'European Lab for Educational Technology

Declaration

This report has been prepared in the context of the ROBOESL project. Where other published and unpublished source materials have been used, these have been acknowledged.

Copyright

© Copyright 2015 - 2017 the ROBOESL Consortium

All rights reserved.

This document is licensed to the public under a **Creative Commons Attribution-NoDerivatives 4.0 International Public License**

Funding Disclaimer

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Table of Contents

Chapter 1: Short description and scenario (O1.1)	3
1.1 The scenario	3
1.2 Connections with subjects	4
Chapter 2: Pedagogical objectives (O1.2)	5
2.1 General objectives	5
2.2 Specific objectives	5
Chapter 3: Suggestions for learning methodologies (O1.3)	6
Chapter 4: Technical guidelines (O1.4)	7
4.1 Building instructions	7
4.2 Illustrative solution	7
4.3 Implementation suggestions	10
4.4 Extensions and variants	11
4.4.1 Variant a: <i>Nextslot</i> , <i>turnleft</i> and <i>checkfree</i> as user blocks [easy]	11
4.4.2 Variant b: Park in the last free slot [medium/difficult]	12
4.4.3 Variant c: Enter slowly [medium]	12
4.4.4 Variant d: Modern automated parking [medium]	12
4.4.5 Variant e: The tale of the three little pigs (with more than one robot) [medium]	13
4.4.6 Variant f: Trolley problem [medium]	13
Chapter 5: Evaluation tools (O1.5)	15

Abstract

This document contains the description of the Curriculum n. 2, entitled ‘Go to park’ which is part of the Intellectual Output 1 (Curricula for 10 exemplary interdisciplinary robotics projects) developed and tested in teacher training courses within the context of the ERASMUS+ ROBOESL project. In this Curriculum we introduce a first form of rotation in the robot which is called ‘swing turn’ because performed maintaining one wheel stopped during the rotation. Checking the emptiness of a parking slot is performed using an ultrasonic sensor, so this curriculum introduces a sensorized robot.

Chapter 1: Short description and scenario (O1.1)



(source: CyberCarteBlanche)

The task is to park the vehicle in the first free one of a sequence of parking slots of the same size. The vehicle moves straight orthogonally with respect to the slots; when it is in the middle point of the current slot, it turns 90 degrees and checks with the ultrasonic sensor if it is free or not, in the first case it moves inside the slot and produces a final sound, otherwise it turns back 90 degrees to move to the next slot; if no slot is found free, the vehicle stops and ‘complains’.

1.1 The scenario

The automotive world is introducing more and more automatisms inside cars offered to the general market. These new features are mainly of assistive nature for drivers but some fully automated cars have already been produced and tested. The teacher is suggested to introduce this experience with a preliminary review of the state-of-the-art car production, highlighting these specific features:

- Automatic parking;
- Obstacle avoidance;
- Driver assistance systems;
- Automobile safety.

In our simplified scenario the robot represents the parking car which moves in front of the sequence of parking slots to look for an empty one (fig. 1).

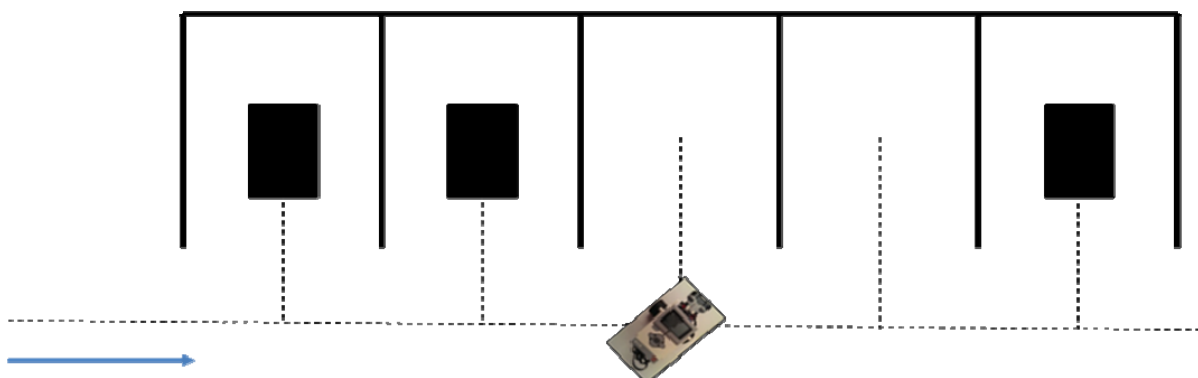


Figure 1. The parking area

Each slot has the same width and the same depth and enough size to accommodate the robot easing its entrance. The robot moves in front of the first slot, stopping in its medium point (some details on this are necessary to provide a correct solution, see below), turns 90 degrees to present its ultrasonic sensor coaxial with the slot axis, detects the possible presence of another car already parked in the slot, and reacts consequently, entering the slot if found empty, or turning 90 degrees back and repeating the process if not.

1.2 Connections with subjects

1. Geographical: on each workday large amounts of people move during the morning from the countryside to the city to work, and come back home in the evening. It is normal that a city administration tries to provide a sufficient number of park-and-ride systems to reduce the number of private cars crossing the city every day. Study some real solutions, analyzing how people can rapidly leave the car parked and reach their workplace with public transportations. Could an automated parking system improve the overall efficiency of the system?
2. Historical: how did the private and public transportations develop in the centuries? Compare the first man and animal power based systems with the mechanized ones, up to the current fast transportations, making evident the difference in energy demands and the aspects related to sustainability and low environmental impact.
3. Philosophical/Ethical: automation moves responsibility from the driver to others involved in the production of more or less automated vehicles. In case of a serious accident, in which proportion the implied blame is, or should be, distributed among the driver, the vehicle manufacturer, the designer of the embedded software etc.?
4. Logical/Philosophical: introduce the so-called ‘trolley problem’ on the autonomous vehicle field. The teacher can introduce this logical/philosophical question to try to have their students improving the technical solution. For instance, adding some external sensors on the rail (internet of things). (There are some different cases and variations of the problem but here you can find a very funny version: <http://www.mcsweeney.net/articles/lesser-known-trolley-problem-variations>). See also below in the variant section.
5. Literature: suggest the students to prepare a sort of movie script where the protagonist is an automated taxi, able to receive vocal commands by customers and to provide a sort of entertainment based on pleasant conversations or answering to music requests or other. Try to conjecture a peculiar situation and a suitable reaction of the car’s artificial intelligence.
6. Science: ask the students to study how a bat uses echolocation to identify obstacles and moving preys (just to start: <http://animals.howstuffworks.com/mammals/bat2.htm>). Try to compare the performance when you have two receptors (like the bat’s ears) and only one (like simple artificial sensors).

Chapter 2: Pedagogical objectives (O1.2)

2.1 General objectives

- To provide students with a stepwise approach for a step by step acquisition of technical skills in using robotic technologies (hardware and software) building on existing knowledge and skills.
- To offer the robotics benefits for all children, especially those at risk of school failure or early school leaving.
- To engage students in STEM related subjects through interaction with the robotics technologies.
- To support self-directed action allowing learners to learn independently.
- To engage students in robotic constructions and problem solving through an interdisciplinary scenario that reflects aspects of real-life problems and situations.
- To align the robotics project to learners' needs and interests through tasks that derive from the initial activity but introduce new levels of complexity and difficulty.

2.2 Specific objectives

More specifically, upon successful implementation of the activities described in this curriculum students will achieve the following objectives:

- Learn to mount the distance/ultrasonic sensor on the tribot for detecting obstacles
- Create the mock-up, the environment into which the robot will operate based on. the scenario
- Make the robot turn around the point of support of the stopped wheel
- Make the robot turn by trial and error experimentations 90 degrees
- Understand through playful experimentation and trial and error the relationship between the rotation of robot and turning angle of the tribot
- To reach the mathematical reasoning underpinning the aforementioned relationship
- Understand the use of the distance/ultrasonic sensor
- Understand the way the command of “switch” operates and implement solutions using this command
- Use of the sound block to introduce sound in the process of parking
- Using a variable for generalizing the solution so that to work for a random number of parking slots

Chapter 3: Suggestions for learning methodologies (O1.3)

This curriculum follows the general methodology common to all the curricula. However, it is tailored to the scenario introduced in the current curriculum. A special worksheet has been designed as a reference and supporting tool. The students are encouraged to work in groups. The teacher acts as a scaffolder and facilitator of the learning process. He/She provides feedback without revealing solutions and probing students through key questions to overcome emerging problems and difficulties. The activity starts with the delivery of the scenario to the students. The teacher in an easy to grasp way elaborates on the scenario. The students are encouraged to discuss the given scenario in groups and to form a general methodology for dealing with the problem. The students are then called to create the mock-up, the environment into which the robot will operate based on the scenario of the activity. The activity progresses with the teacher explaining the way the distance/ultrasonic sensor can be mount on the tribot.

The students are then encouraged to work with the tribot following the worksheet. They are called to explore through trial and error ways of making the robot turn around the point of support of the stopped wheel. Again, through trial and error and playful exploration the students are called to make the robot turn 90 degrees. Students may keep notes regarding the rotation values that they tried out. The teacher will support the students during this process and in the end will encourage them to discover the mathematical solution underpinning the given scenario based on the results of the experimentation/trial and error.

The teacher provides all the necessary information regarding the way the distance/ultrasonic sensor operates. This stage is followed by practical experimentation by the students. The activity is concluded with the use of the command switch. The teacher introduces the main functionality of this command to the students through easy to grasp examples. The students then will be called to introduce the command switch in the context of the given problem as part of the solution.

Optionally, after the completion of the activity the students are also encouraged to explore a more general solution that will not be dependent on the 5 slots but it will work for a random number of slots. For this task the role of the variable should be exploited.

A. The role of the students

Students first discuss the problem through a free dialogue in their group and after that they devise an action plan to solve it. They work in groups following the worksheet and the discrete feedback they receive from the teacher. Students may extend their work to variants suggested by the teacher or devised by students themselves. First, they find solutions making trial and error experimentations. In the end they are supported to find a mathematical solution to the same problem. The final solutions of the groups are presented in the class, are discussed and evaluated with students reflecting with critical mind on their work, expressing their views and recording their experiences in a diary or questionnaire.

B. The role of the teacher

The teacher in this constructivist theoretical framework like that described above does not function as an intellectual “authority” that transfers ready knowledge to students but rather acts as an organizer, coordinator and facilitator of learning for students. S/he organizes the learning environment, raises the questions / problems to be solved through a worksheet, offers hardware and software necessary for students’ work, discreetly helps where and when necessary, encourages students to work with creativity, imagination and independence and finally organizes the evaluation of the activity in collaboration with students.

Chapter 4: Technical guidelines (O1.4)

4.1 Building instructions

For this experience we can use the same *tribot* structure used for the previous experience on which we have mounted an ultrasonic sensor. The way and the position we chose to mount this sensor is not of relevant importance, but we must take care of putting it on the same main axis of the brick (and of the whole robot) so that it can measure the distance with respect to objects present in front of the robot (fig. 2). Moreover, as we will see soon, we want to make the robot turn in a simply way: the caster ball in the *tribot* configuration permits this turning with limited friction.

The scenario can be easily built by the students, either drawing on a large sheet of paper the layout of fig. 1, or recreating the scenario with separation items like Lego, wood or other plastic pieces. The only important details are to create slots of the same size and equally distributed in space and to have objects to be put inside the slots to 'occupy' them and sufficiently high to be 'seen' by the ultrasonic sensor.



Figure 2. The ultrasonic sensor mounted

4.2 Illustrative solution

One of the purpose of this experience is to introduce a new motion command, the Move Tank command (fig. 3). Some of its parameters (power, duration, brake) have the same meaning of the Move Steering command. The main difference is that in the Move Tank command the two motors are singularly controlled with independent powers. This means that the two wheels can have different angular speeds: this leads to make the robot turn.

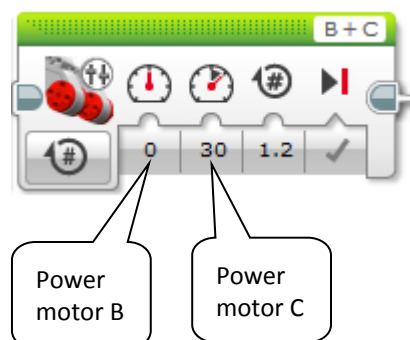


Figure 3. The Move Tank command

Because in this experience we are interested in turning 90 degrees, the teacher is suggested to introduce a simpler method, and usually more straightforward for the student, given by one stopped wheel and one moving during the rotation (fig. 4). This type of turn is also called ‘swing turn’. More complex turnings will be presented in the following experiences.

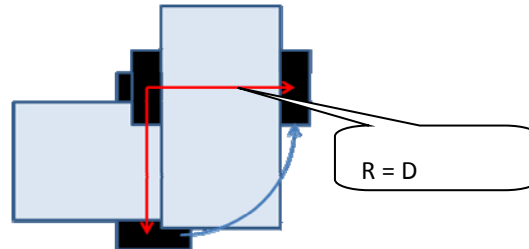


Figure 4. Swing turn

Like in the previous experience the teacher could leave the students to find experimentally by try-and-error the amount of degrees to be spanned by the moving wheel to produce a turn of 90 degrees of the robot around the point of support of the stopped wheel. For example, by means of a protractor the students could evaluate the number of turning degrees of the robot corresponding to an entire rotation (or an integer number of rotations) of the moving wheel and calculate by proportion the amount of wheel rotation to cause a turn of 90 degrees. Fig. 4 suggests the applicable geometric model where the radius R corresponds to the distance D between the two wheels (between their points of support, to be precise). Say r the radius of the wheels, α the angle to be set for the wheel, for the arc of $\theta_g = 90$ degrees (or $\theta = \pi/2$ radians) ‘drawn’ by the robot it holds:

$$\theta * R = (\pi/2)*R = \alpha * r \quad (\text{in radians}) \quad \Rightarrow$$

$$\alpha = (\pi/2)*R / r \quad (\text{in radians}) \quad \alpha = 90 * R / r \quad (\text{in degrees})$$

Once discovered how to turn and how to set with precision the necessary wheel angle, it is now possible to design the program. Again we can recognize that the task is the repetition of the same subsequence of actions: a straight motion, a rotation of 90 degrees, a checking of the emptiness of the slot and, consequently, the decision to enter the slot or to turn back and repeat the sequence.

In this case we want to make the program more general, flexible with respect to the number of slots, the wheel radius and the distance between wheels which are all configuration parameters in our scenario. For representing each of these parameters, we can use a *constant* (fig 5a), which is essentially a container for a value which must be set at the beginning of the program, from that point after only read through data wires; because constants are unnamed, to improve readability you could prefer to use a named *variable* to be written only once at the beginning, and never modified, as if it were a named constant. In fig. 5b an example with N representing the number of slots: if you change the number of slots in the setup, it suffices to change the initial value of the ‘constant’ N .



Figure 5. a) A constant value – b) A read-only variable

After each turn of 90 degrees, there is a decision moment: such decision is based on the measure returned by the ultrasonic sensor which, as officially declared, within the range 3..250 cm and an angle of about 20 degrees, can detect an object and measure its distance with a precision of +/- 1 cm. This gives the occasion to introduce in depth the Switch command (fig. 6). The condition on which the decision is taken is represented by the option chosen in the 'premise' of the block, in our case the measure returned by the ultrasonic sensor. The measure is performed any time the command is executed and it is compared with a threshold; the result of this comparison is the logical value on which the switch acts: if true, the upper subsequence is executed, otherwise the lower one is executed, being these two parts alternative.

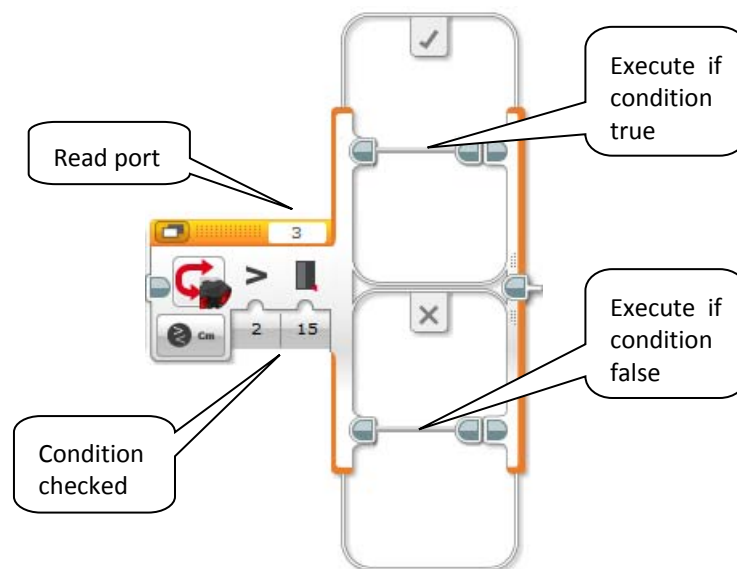
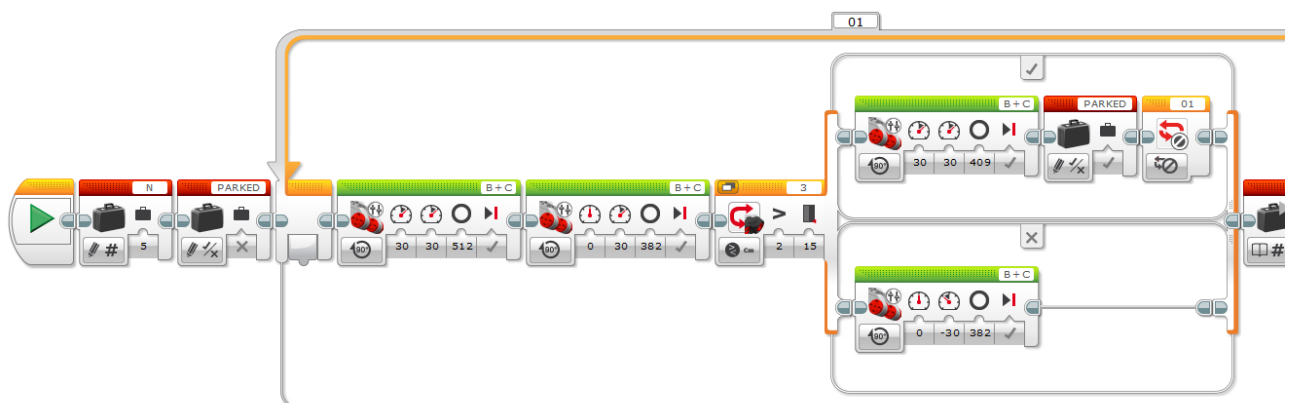


Figure 6. The Switch command

Now we can present in fig. 7 the entire program which includes the production of the final sounds.



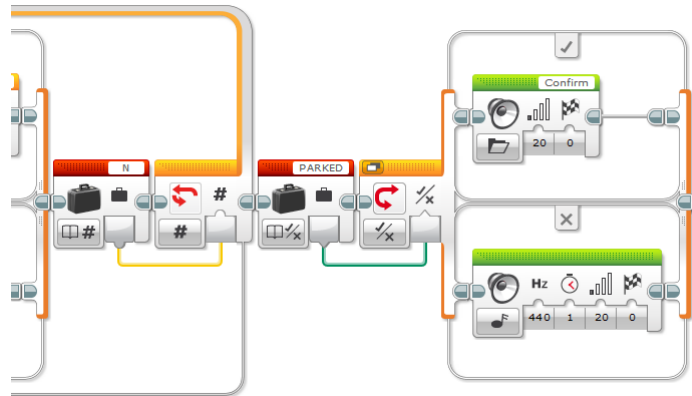


Figure 7. The parking program

The variable N is set to the number of slots and used to provide the repetition factor of the loop block; $PARKED$ is used to signal at the end whether the parking was successful or not, in order to choose the appropriate sound. All the calculations for the degree parameters are done here assuming a slot width of 25 cm, a space of 20 cm to enter a slot, a wheel radius $r=2.8$ cm and a distance between wheels $D=11.9$ cm. For example, for the straight-line motion, it holds:

$$\text{degrees} = (180/\pi) * \text{space}/r = 57.3 * 25 / 2.8 = 512$$

whereas for turning:

$$\text{degrees} = 90 * 11.9 / 2.8 = 382$$

There is a detail which the teacher could highlight after the first executions of the program. Due to geometrical aspects of the chosen type of turning, the position of the robot after the turning is $R/2=D/2$ units ahead the position reached after the straight motion: this means that, while all the straight motions must have the length S equal to the width of the slots, the initial position (referred to the wheel axis) must be $R/2+S$ unit from the axis of the first slot in order to have always the robot aligned in front of a slot after a turning (compare with fig. 1).

If you want to make the program even more general, you can add a variable r representing the wheel radius, and D representing the distance between wheels. The straight motion is now performed by the code of fig. 8a (slot width again 25 cm), whereas the turning code for 90 degrees is shown in fig. 8b.

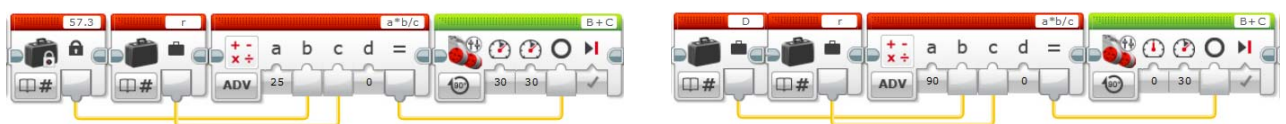


Figure 8. a) Parameterized straight motion – b) Parameterized turn

4.3 Implementation suggestions

This experience can motivate the students starting with the construction of the scenario of the parking area. This does not require neither complicated procedures nor specific materials hard to be found, only a little care in setting distances. Because the decision about the emptiness of a slot is not so sensitive to the position of the obstacle which possibly fills the slot, it is only required to establish a suitable threshold to reliably distinguish the two conditions (empty or full).

In spite of its relative simplicity, one of the less straightforward concepts to be introduced is the ‘variable’. This is partly due to the meaning learnt in the context of mathematics where a variable is not exactly the writable container that should be thought in the context of a programming language. This first occasion of using variables is reduced to the constant N, something written once and read many times. Justifying its introduction to make the program less dependent to the number of slots, i.e. avoiding to change the program in several points if this parameter changes, seems enough understandable and preceding the use of real variables in more complex programs.

In the first experience students learnt to move the robot by controlling its motors; making now the robot turn is not a big change, it is still a matter of suitably applying a proportion, after some preliminary experiments. Nonetheless the students are asked to think about turning as a general behavior compliant with a mathematical/geometrical model, a bit more complex than the model used for the straight line motion. Step by step they are constructively becoming aware of the rules which govern a robot navigation: this process of awareness will be completed in the following experience.

4.4 Extensions and variants

4.4.1 Variant a: *Nextslot*, *turnleft* and *checkfree* as user blocks [easy]

Similarly as the *Track* user block presented in one variant of the first experience, here we want the students to make three user blocks representing respectively the straight line motion, the swing turn of k degrees to the left (counterclockwise), the verification of emptiness of the slot which must return a logical value. The realization is at this point rather simple (fig. 9, 10, 11). Notice that you can also realize a *turnright* user block similar to *turnleft* but with the two motor powers swapped.

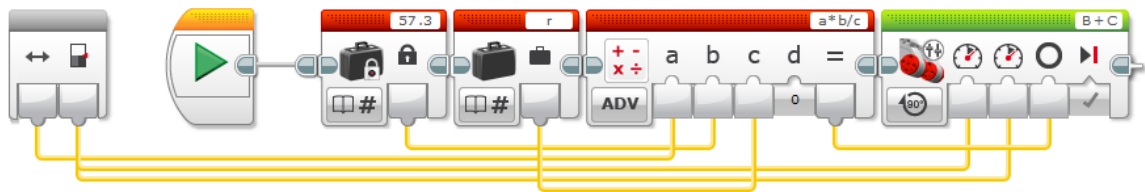


Figure 9. Nextslot user block

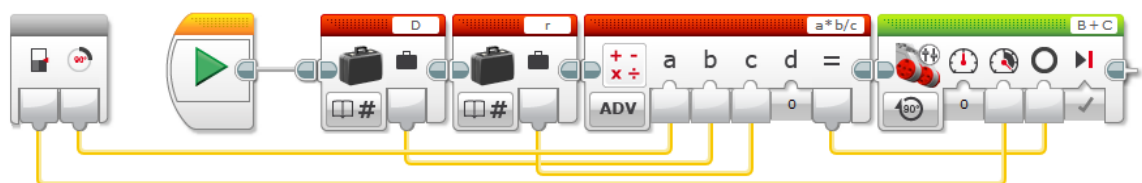


Figure 10. Turnleft user block



Figure 11. Checkfree user block

4.4.2 Variant b: Park in the last free slot [medium/difficult]

In this case the robot must make the verification of emptiness in all the slots, storing the position of the last empty slot in a variable LAST initialized to 0. When it reaches the last slot, the content of the variable is used to evaluate of how many slots the robot must be retracted in order to reach the last empty one. For example, say N the number of slots, starting from index 1 to N, if i is the final content of LAST, the robot has to be retracted for $N-i$ slots, this difference being able to be equal to 0 if the last slot is empty, and $N+1$ if no empty slot is found; in this last case, the robot should stop and 'complain'. N could be known from the beginning or evaluated run-time recognizing the end of the slot sequence, for example putting an obstacle on the straight line near after the last slot.

This variant has another further extension: instead of storing the index of the last empty slot in a single variable, during the motion the robot could store the state of each slot in one successive element of a N-items array, for example 1 or true if it is empty, 0 or false if full. At the end the robot would perform a linear search on the array from the last element backward to find the index of the last empty slot and retract consequently. This is an excuse to introduce the array data structure if it is of interest for the teacher.

4.4.3 Variant c: Enter slowly [medium]

The depth of the slot is not known: the slot is terminated by an obstacle, for example a wall (a box or a piece of wood or similar). Using the ultrasonic sensor, the robot must enter the slot at decreasing speed until it reaches a minimum distance from the bottom of the slot and a minimum speed, after that it stops. Speed, i.e. power, is now related to the distance from the bottom, more precisely the value of power is a direct function of the measured distance (for example, a linear function with a positive coefficient of the distance which represents the independent variable of the function). Evaluating analytically the motion that comes out from this control law is beyond the scope of this text (being theoretically the solution of a differential equation).

4.4.4 Variant d: Modern automated parking [medium]

Many actually produced cars are equipped with automated systems able to assist the driver during parking or even to fully control the car to safely park. These systems use a network of sensors which gives a sufficiently complete view of the surrounding environment to obtain the maximum of precision of parking or declare its unfeasibility (fig. 12).

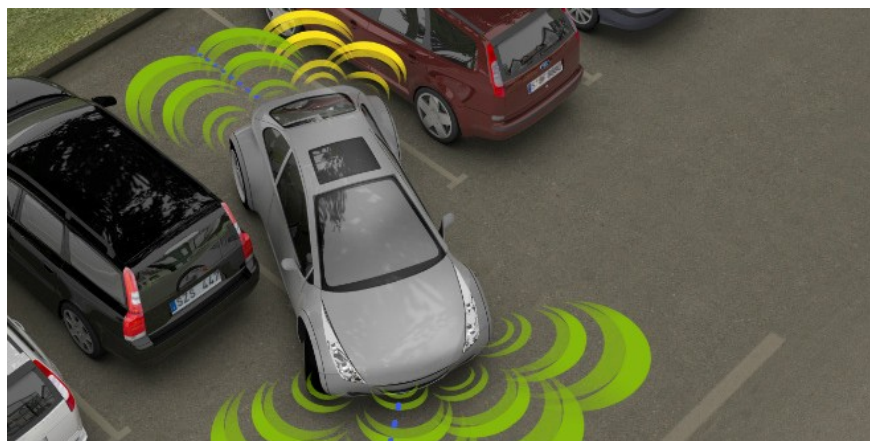


Figure 12. Full automated parking (source: davincitech.it, valeo.com)

In this variant of the experience we want to emulate an automated parking using just one ultrasonic sensor but mounted so that it can detect objects put on one side of the car. The sequence of actions we want to program is the following (fig. 13):

- Moving on a straight line, the car pass beyond the open space between the two boxes which delimit the parking slot to align with the beginning of the second slot;

- It retracts a bit;
- It turns 45 degrees to put itself diagonally with respect the slot axis;
- It enters within the space;
- It realigns to the slot axis turning 45 degrees in the opposite direction than before.

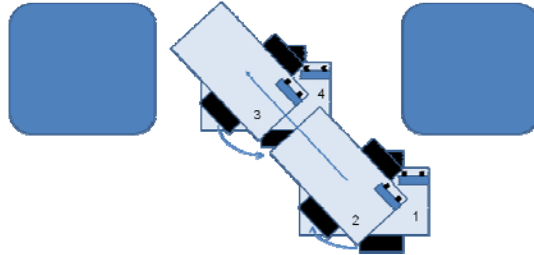


Figure 13. Automated parking emulation

4.4.5 Variant e: The tale of the three little pigs (with more than one robot) [medium]

We want to emulate the famous tale: prepare a scenario with three areas representing the three different houses (like fig. 14). Ask the students to translate the story of the wolf in a sequence of actions involving more robots, one for each character, possibly in form of a storyboard. Suggest the students to limit the motions to the ones they know, straight-line and turning of 90 degrees.



Figure 14. The three little pigs tale emulation

4.4.6 Variant f: Trolley problem [medium]

First present the general problem and then make the students prepare a scenario with a decision point between two orthogonal continuations (fig. 15). When the robot reaches this point, it can know how many potential 'victims' are ahead, a greater number if there is a high object detected by the ultrasonic sensor, a lower if it is not present (or not visible). Then it chooses either to proceed or to turn to the other line; this choice is made applying a certain probability in favor of one of the two options, to choose the line with one man or leave the robot to go straight (no intervention). Discuss with the student their proposals for this probability.

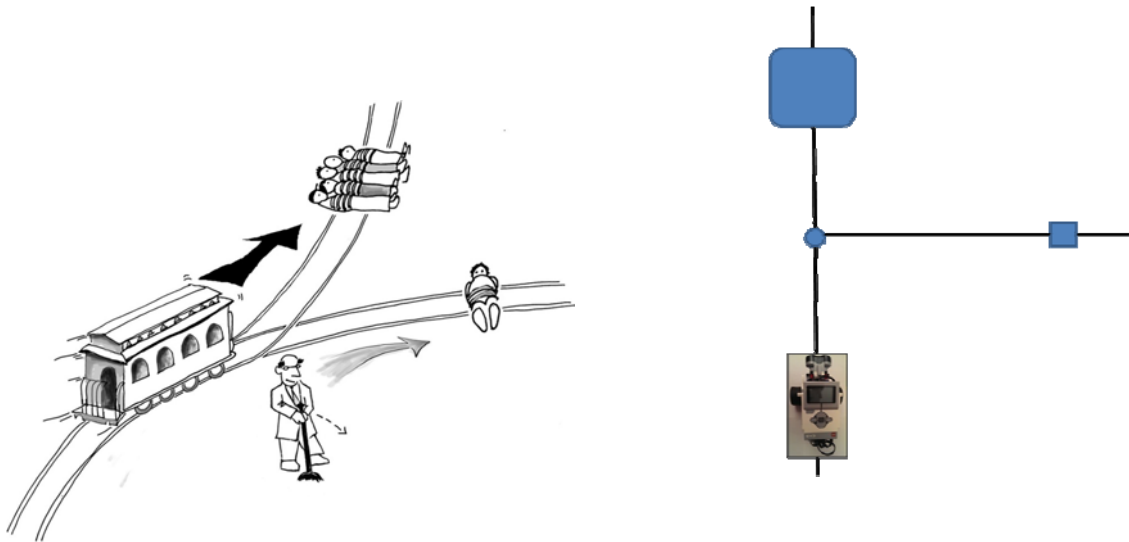


Figure 15. The trolley problem emulation

In order to apply a choice with some distribution of probability, you must use the **Random** command (fig. 16). It gives back a random integer number chosen in the indicated range of values, each number having equal probability (the so called ‘uniform discrete distribution’). If the range is between m and n , you have $(n-m+1)$ different integer values, each one with probability $1/(n-m+1)$. If you have only two choices A and B and you want that the choice A is generated with probability p_A and B with probability $p_B = 1-p_A$, one simple method to obtain this type of generation is to choose m and n and an intermediate threshold t ($m < t < n$) so that it holds:

$$p_A = (t-m+1)/(n-m+1)$$

Then choose A if the generated number r is $m \leq r \leq t$, otherwise choose B.

For example, if you want $p_A = 0.3$, you can choose $m=1$, $n=10$, $t=3 \Rightarrow p_A = (3-1+1)/(10-1+1) = 0.3$,

being A associated with numbers 1..3, B with numbers 4..10. You can obtain the same effect using the logical option of the Random command setting the probability of the true output to the required value.

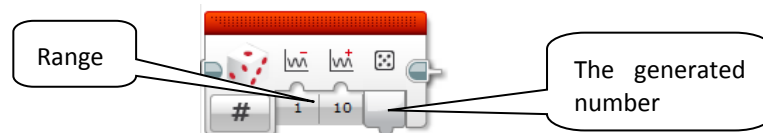


Figure 16. The Random command

Chapter 5: Evaluation tools (O1.5)

Use the rubric below to evaluate your students' achievement in each specific objective of this curriculum.

Name of student (or group of students):

upon completion of the activities described in this curriculum students achieved the following objectives	Evaluation score 0 = not attempted 1 = attempted without success 2 = partial success 3 = completed with teacher's help 4 = completed without teacher's help
Built correctly a ultrasonic sensor equipped tribot	
Translated the scenario of the parking area into a realistic representation	
Properly used an ultrasonic sensor	
Discovered and properly used the "move tank" command to make the robot swing turn	
Discovered and properly used the "move tank" command to make the robot swing turn	
Properly programmed a swing turning motion	
Discovered and properly used the "switch" command	
Discovered and properly used the "sound" command	
Instructed the robot to autonomously park in the first free slot	
Properly defined user block(s) for the variant a)	
Discovered a mathematics-based solution for making the robot turn correctly	