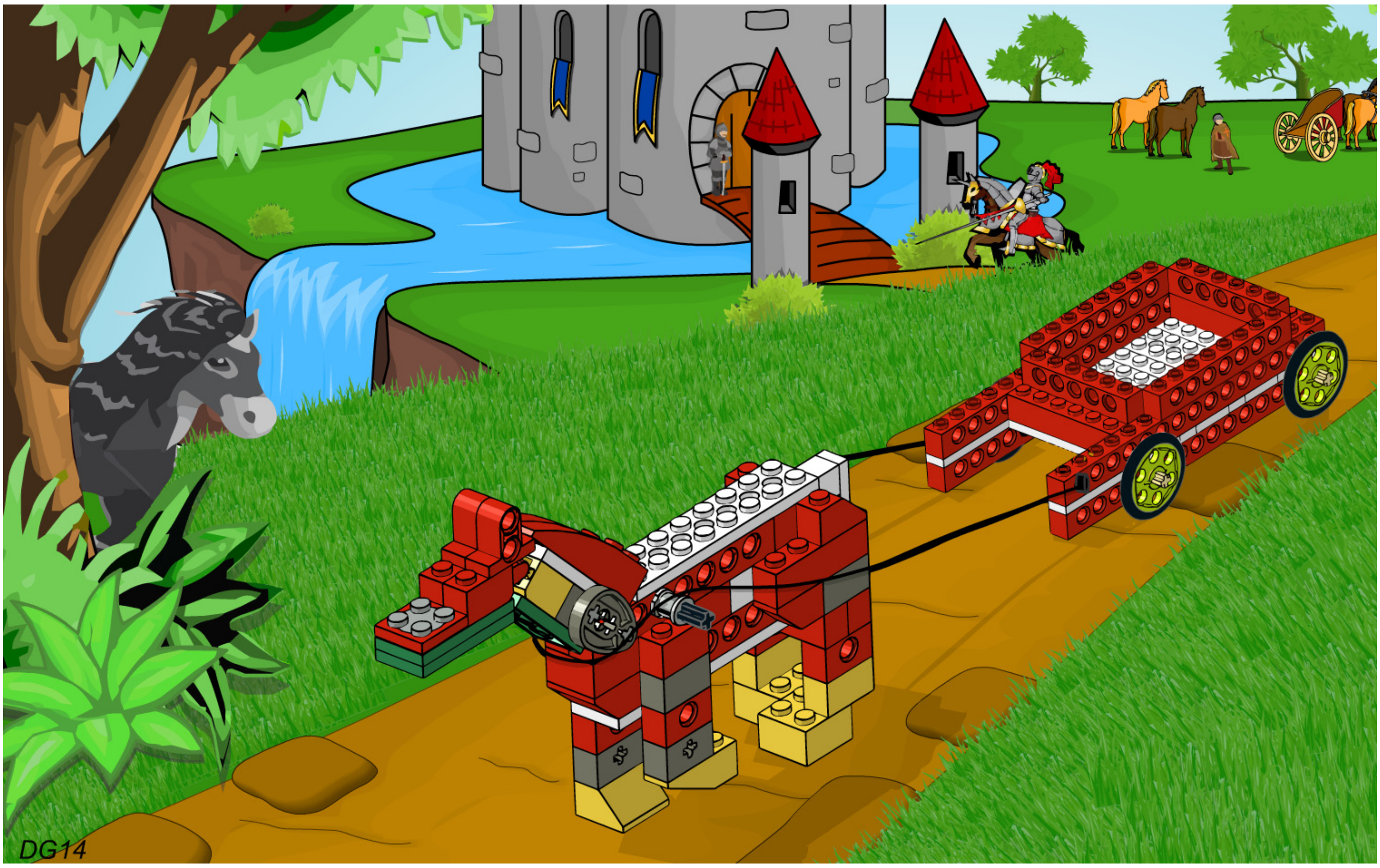


Robotics with LEGO® WeDo (VII)

An introduction to robotics for the young with LEGO® WeDo

By Diego Gálvez

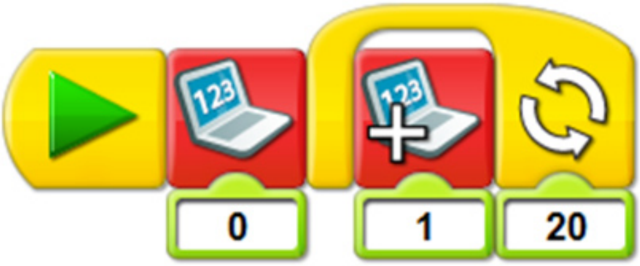
In this part we will continue working with some aspects of the send and receive message blocks

Double Loop

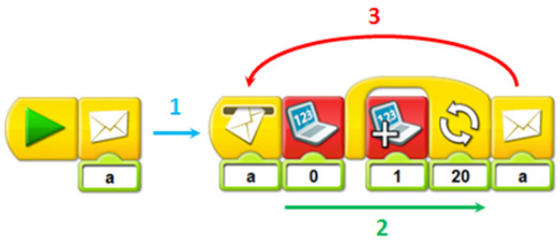
The WeDo software contains a repeat block that allows us to program loops. But what if I want to use a loop inside another?

Example

We will make a program that counts up to 20 and then starts over again, counting to 20 each time. The first thing to do is to create a program that is going to count from 0 to 20.



If we run this we will see that it indeed counts from 0 to 20, after that it stops. If we want to repeat the program we have to click on the start block.



To automate this process we are going to use the message block.

1. Message “a” is sent.
2. Message “a” is received and the program is executed.
3. The program ends and the message “a” is sent again, starting the program all over.

WORKING WITH CASES

In later parts this will be the function of the “send message” and “receive messages” blocks we will use the most.

What does working with cases mean?

Well, imagine the following everyday example. Depending on the hour of the day you have breakfast, lunch or dinner. From a programming point of view the hour of the day would be our variable, and breakfast, lunch or dinner the action to take depending on the value of that variable.

Before we continue, we need to know what a variable is in programming.

Variables

In programming, variables are spaces in memory that are reserved and which, as the name indicates, their contents can change during the execution of the program. A variable corresponds to a reserved area in the main memory of the computer.

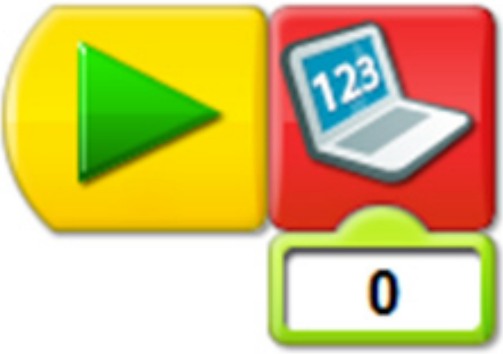
Variables in WeDo

In the case of the WeDo software we can only work with a single variable which is the “number on screen”. The add, subtract, multiply and divide block allow us to change that “number on screen” and use it as a variable that indicates which case should be executed.

Example

Initialising a variable

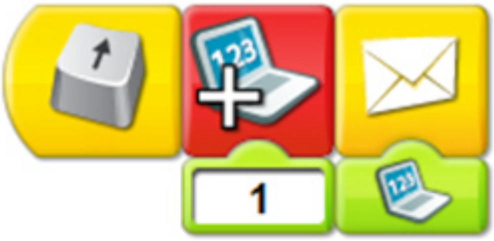
In some programs you need to initialise the variable, which means you give it a starting value. In this case that value is zero.

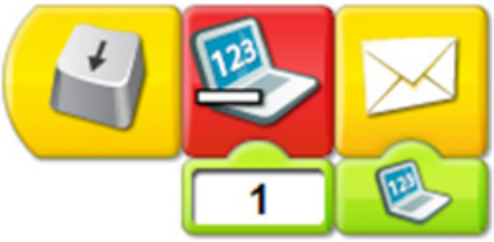


This small program runs at the beginning to verify that the variable starts with the desired value.

Changing a variable

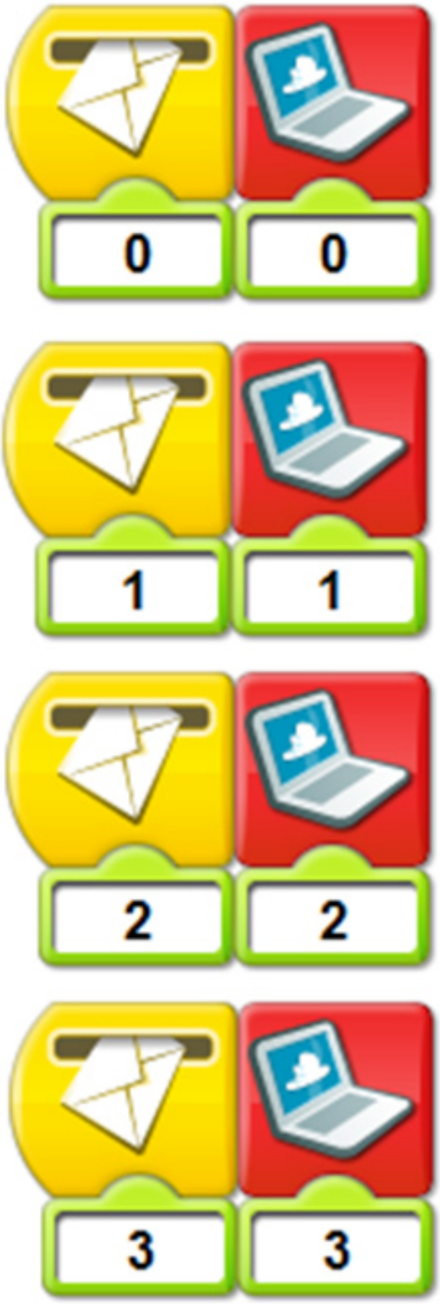
The next step is defining how the variable will change. This can correspond to the reading of a sensor or to the program adding, subtracting, dividing or multiplying another value. In this example we are going to use the keyboard block: when arrow up is pressed the value will increase one unit, when arrow down is pressed it will decrease one unit.



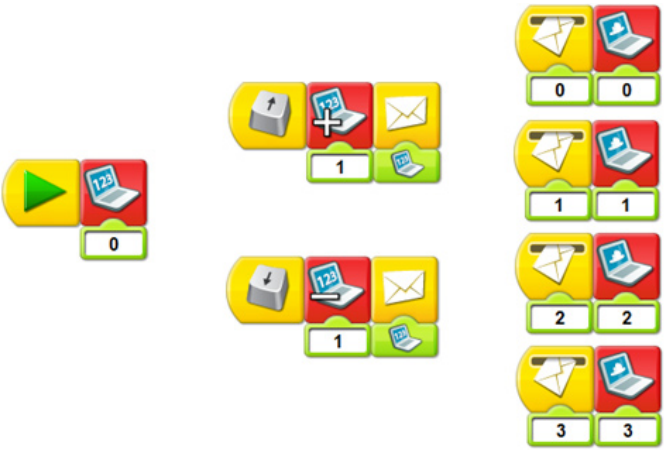


Adding cases

Finally we need to add cases. That is, the action that should occur depending on the value of the variable. In this example there are 4 cases.



The final program looks like this:



Let’s try changing the number on the screen by pressing arrow up or down to see how the background changes.

As we have seen, the send and receive message blocks can be used to create more complex programs.

In the next part I will explain how to create and program a line follower with the WeDo set. For the program we will use these “send message” and “receive message” blocks.

On the website notjustbricks.blogspot.com you will find multimedia materials (images and videos) of the creations of the author, some of which come with building instructions.
#

NOT JUST BRICKS
NO SON SOLO LADRILLOS