



LEGO® WeDo (VI)

Programming in Scratch

By Edwar Romero

Cover image by Osvaldo Romero

We are still here with our world domination plan! We are getting close, just one more LEGO® brick, but give me the programmable WeDo™ bricks! Yes, we know that we can program those bricks with Scratch as we please. For the time being, we will continue with the basics until we are able to release our inner genius. We will learn enough so we will be able to create a Terminator-like robot, but decorated with a thousand color bricks!

In summary, we have compared codes elaborated in WeDo against codes developed using Scratch following the same approaches: from turning the motor and the different ways to do it to how to program more sophisticated versions.

We have discussed the Amazing Mechanisms: the Dancing Birds, the Smart Spinner and the Drumming Monkey. We have talked about the Wild Animals: the Hungry Alligator, the Roaring Lion and the Flying Bird. So, we are about half way through the basic WeDo models. Now we continue with the Play Soccer activities: the Goal Kicker, the Goal Keeper and the Cheerful Fans.

Let's start with the Goal Kicker. The WeDo program begins with the play button to turn the motor on counter clockwise, but just for a short time (only 0.2 seconds or 2 in WeDo, 10 corresponds to 1 second). This is enough to send the ball to the goal against the other team.



The code in Scratch is quite similar. You need to use the equivalent of the play button (“when the flag is clicked”) under the Control Menu. You also need the motor commands under the Motion menu, “Motor direction this way” and “motor on for ___ seconds” with the time as 0.2 seconds.



That wasn't difficult, was it? It can be more interesting later on. For instance, why not use the distance sensor to detect when the ball is in front of the robotic leg. That way the robot can shoot straight ahead to the goal line.

This can be done in the WeDo software with the waiting block (the hourglass icon) attached to the distance sensor (the one with eyes) so when the ball is in front of it, the robot will kick it.



In order to do the same in Scratch, you need to know where to look for the commands since they are kind of hidden. You need to use the “wait until” block (the hourglass equivalent) under the Control menu. We know the distance sensor is required, but we need another block first. Scratch will not allow us to connect this command until we add a comparison block from the Operators menu (the light green commands). On that menu, there is the “_<_” (less than) code. Under the sensors menu, close to the bottom is the “__ sensor value”. From the drop down menu you need to choose distance and you also need to write 2 on the right empty cell. Quite a journey, isn't it?



If we were able to configure a robotic leg to kick the unfortunate soul crossing in front of it with a mega kick, then we can dream about a sophisticated security system. So there is no need to settle for basic alarms, right now we can kick intruders to our property with our coding ability and our LEGO sets. One step closer to conquering the world!

In a soccer game we also have a goal keeper to face the goal kicker. Since not many people like to be the goalie, what better than having a robotic friend to block the shots and to run between the posts. Penalties now look more interesting!

For our goal keeper to be really effective between the posts, it needs to move really fast without the other team being able to forecast the moves. For that we need to use random inputs using the dice in the WeDo software.

We need to use the start button (the play icon) and then the repeat block for the goalie to continue moving like crazy. We need to assign one direction for the motor, then to wait a random time (using the dice), turn the motor the other way, then wait another random time before moving the other way, and to start all over again.



Coding in Scratch is quite similar. We need to look for the start block (“when the flag is clicked” command) and the repeat forever block. Those are under the Control menu. The wait command (“wait ___ secs”) is also found there. The motor commands are found under the Motion menu for the motor to turn this way and to turn in reverse. You need to look for the “pick random ___ to ___ secs” code under the Operators menu. You need to fill out the numbers 1 and 3 on the empty cells.



However, who is taking note on the score? Is not easier to develop a program to do it automatically rather than manually? Why not coding it for to the computer to carry the tabs. You can do it with the WeDo software in another piece of code running alongside the goal keeper. We need the Display block (the red icons) configured to start with the number zero (to start the score using a white background). A repeat block is also

required, the wait (hourglass icon) with the distance sensor attached (to detect if a ball crossed the goal line), another display block (add to display block to add 1 every time an annotation is detected), wait for half a second (the number 5 on the image), and repeat the procedure all over.



WeDo will show an image similar to the one shown below with the score of the game.



The programming in Scratch follows the same approach as the WeDo code (a second program alongside the goal keeper code). You need the start button (“when the flag is clicked”) and a score counter starting from zero. For this, you need to create a new variable called score. That needs to be done under the Variables menu (Make a variable and click the checkbox to make it visible on the white canvas). The first time you enter the menu it will be empty. After that, you need to use the “set ___ to ___” block, choose the variable you created from the drop down menu and add a zero. Under the Control menu you need to choose the forever block, the “wait until ___” and the “wait ___ secs”. On the Sensors menu you need to choose the “___ sensor value” block, find the distance sensor on the drop down menu. You need also to add a comparison block, the less than (“___<___”) command from the Operators menu. You can test variations of this number for better results (50 was found adequate for this part as in the image). You need to program the wait block for half a second, otherwise it may count the same annotation multiple times. Last, but not least, you need to choose the “change ___ by ___” block from the Variables menu and select the increment for the last command.



This program will show a counter on Scratch similar to the image shown below with the score.



We are almost done, next are the Cheerful Fans to complete the soccer match. The program developed in WeDo is as shown below. The motor is turned on to run one way, a sound is played, wait for a bit, another sound is played, wait a bit more and turn off the motor.



This program can be written in Scratch as the figure shown below. Yet again, the more complicated part is finding the sounds you like. For this you need to import them from the Scratch sound libraries (Sounds tab on the programming area) or to record one for your beloved soccer team.



If we are capable of programming robots for a soccer game, including cheerful fans and multi colored bricks, I think we are capable of bigger things. It looks like world domination is getting closer, but let me first finish my robotic soccer game.

That's all for now folks, stay tuned for the advanced programming of these creations in the next issues of HispaBrick Magazine®. You can find more information, and building and programming instructions for the designs presented here and much more at:

www.wedobots.com
www.facebook.com/wedorobots
 #



wedobots
 The unofficial blog for LEGO® WeDo designs