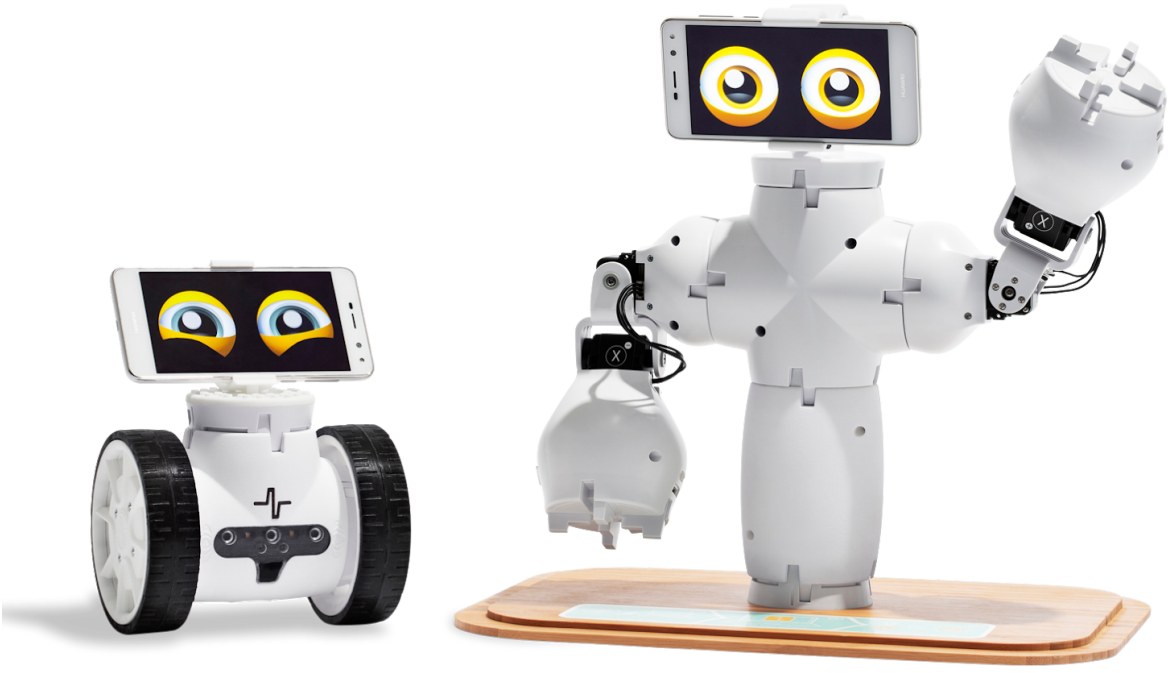
Fable

Getting Started



Vers. 1.5
Updated: 03-07-2019

Content

1.Install the application 3

2.a Fable Joint - unpack the robot 3

2.b Fable Spin - unpack the robot 5

3. Start programming Fable 6

4. Move from 90 to -90 degrees 9

5. Introduction to loop 10

6. Program colors 11

7. Learn about conditions 14

8. Control Fable Spin 16

9. Fable Spin drive and turn 18

10. Make a surveillance robot 19

11. Make a graph 21

12. Log data with Fable 23

13. Fable Face 24

14. Control Fable with smartphone 26

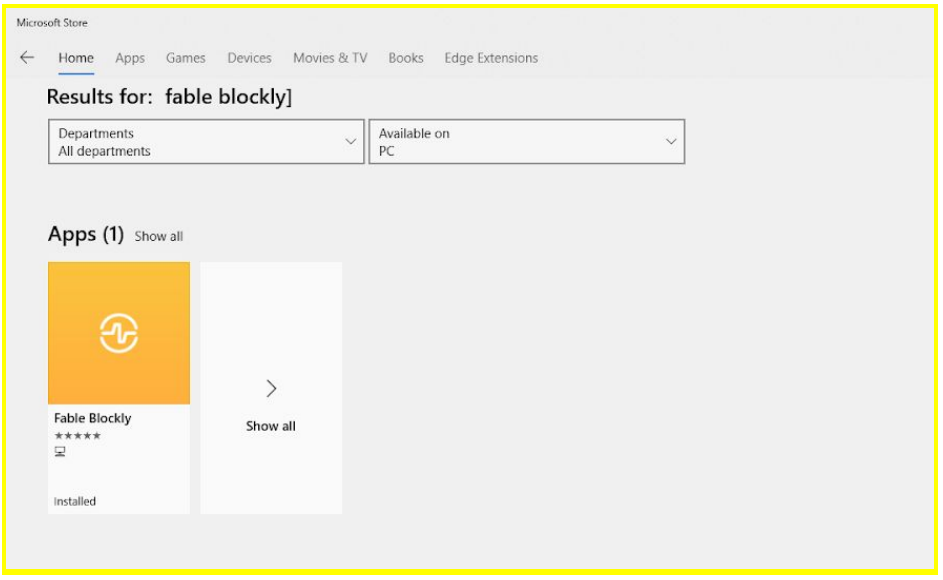
16. Refine the program 27

17. Make a variable 28

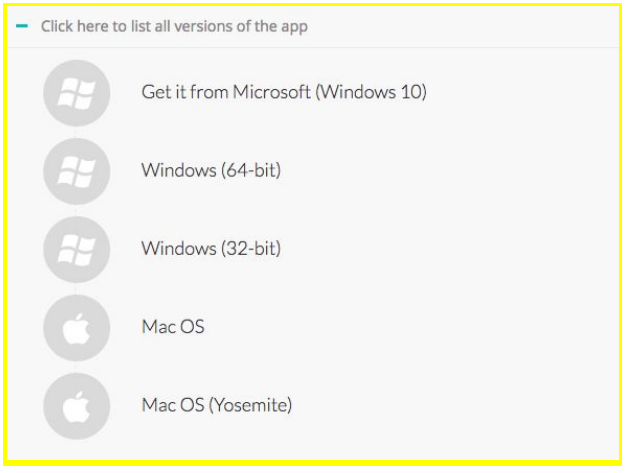
18. Save, open and examples 30

1. Install the application

If you are running Windows 10 you should go to Microsoft Store and search for *Fable Blockly* and then choose download.



If you are running Windows 7, Windows 8 or Mac OS you should open the following webpage - shaperobotics.com/download - and download the app for the respective operating system.



Windows 7 and **8** will also need a hub driver – so you should also download that (find instructions on this at: shaperobotics.com/pages/faq/).
If you are using Mac OS you should be aware if you have updated your operating system with the latest updates.

2.a Fable Joint - unpack the robot

First take the lid off the box.

Flip the lid so that the stand is facing upward. Take out the Joint module.

Next, insert the Joint module into the stand with its largest end at the bottom, so that the Fable logo is facing you.

Turn on the Joint module by sliding the switch on the back. Notice that the Joint module has a “name” - this one is called PCA. Remember your Joint module’s name.

The Joint module will now light up in a certain color as shown in the picture below:

Now it’s time to take out a Hub and a USB cable. Insert one end of the cable into the Hub and the other into the computer’s USB port (Note: The Joint module does not need to be connected to any cable).

Once connected, the Hub will light up. This should be the same color as the Joint module. If it isn’t, press either the Hub or the joint module until they have the same color.

2.b Fable Spin - unpack the robot

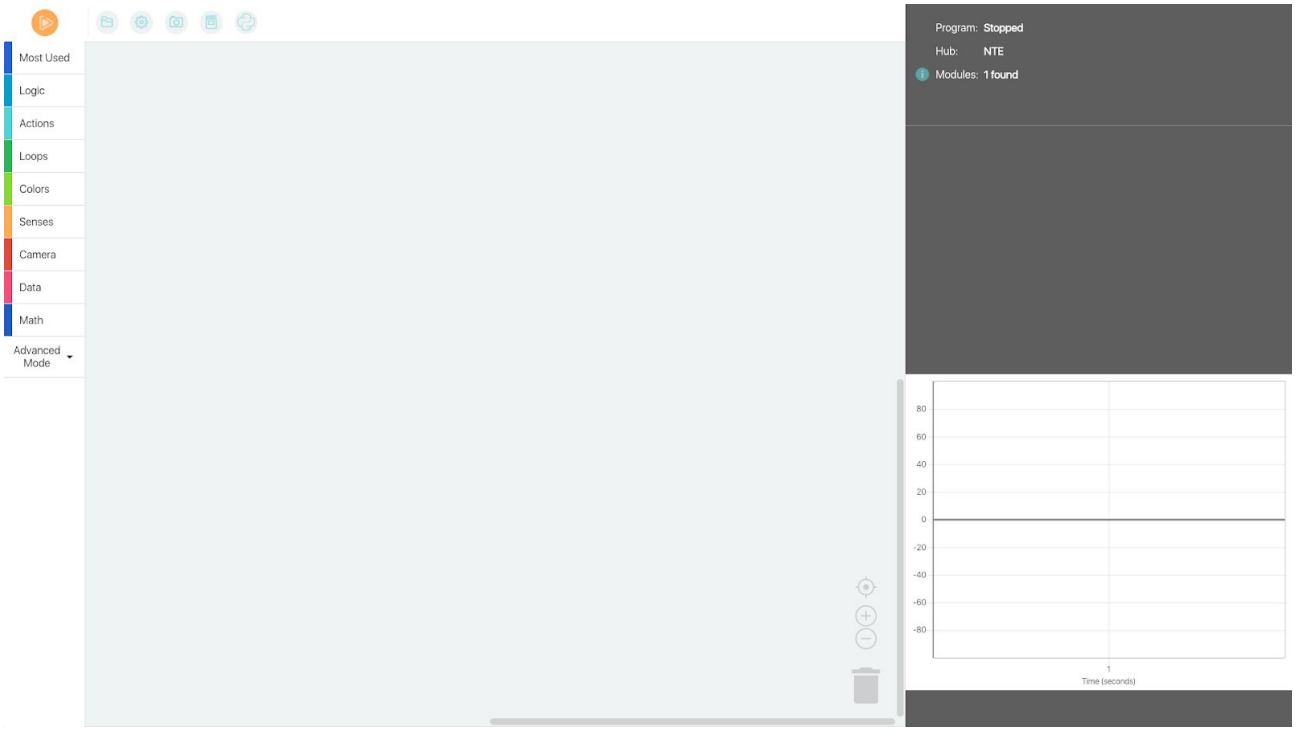
Take out the Fable Spin and mount the castor wheel behind the robot.

Turn on the Spin by clicking the Shape Robotics icon (note: On Fable Spin you just click this button to turn it on). You will find the ID of the Spin on the opposite site:

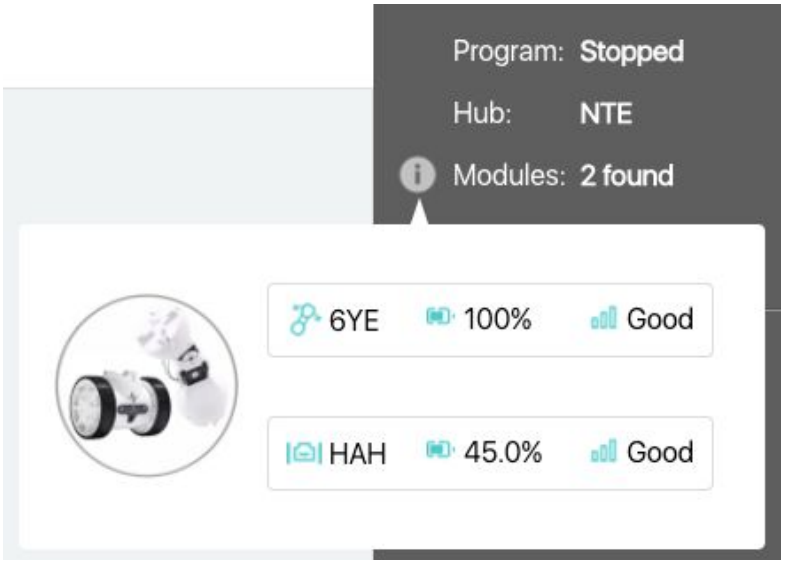
IMPORTANT: NEVER RUN THE SPIN ON THE TABLE WHEN IT IS MOUNTED WITH WHEELS.

3. Start programming Fable

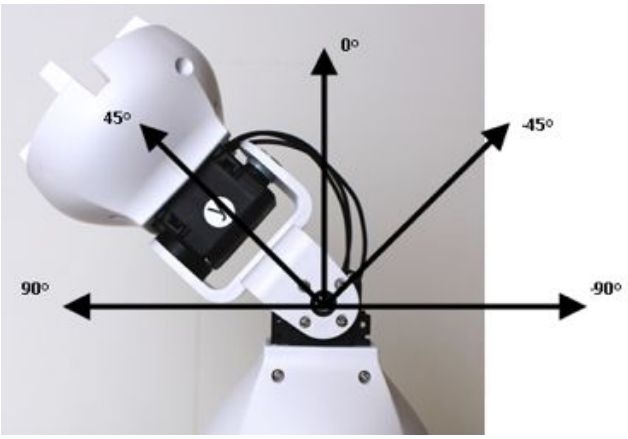
Open Fable Blockly on your computer.



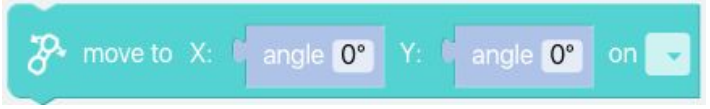
On your computer, check whether the software has "found" your Hub. You will find the connected modules under *modules*, and if you click the little *"i"* you will find the name of the module, the battery level and how good the connection is. If it does not work consult this guide on our Help/FAQ page.



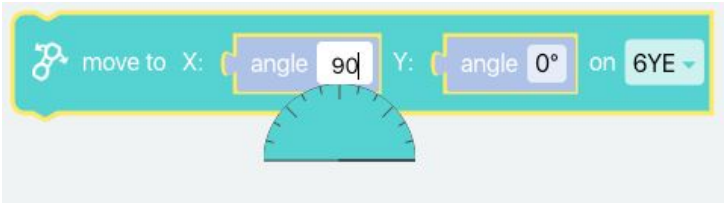
The Joint module has 2 servos, X and Y, which can move from 90 to -90 degrees. If you have connected the Joint module as shown in chapter 2.a, the X servo should swing left at 90 degrees and right at -90. The Y servo should swing towards you at 90 degrees and away from you at -90.



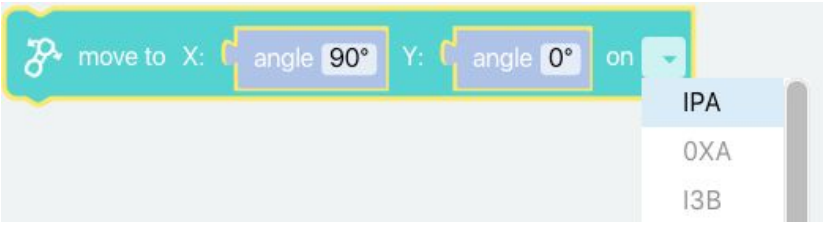
Click on **Actions** and drag the block called ***move to X*** to the right (into the white area):



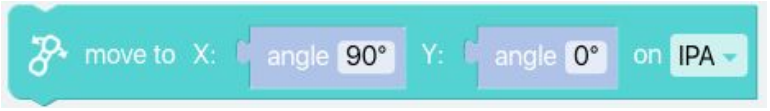
Try changing the angle of the X servo to 90 degrees (we will leave Y alone for now):



Remember to change the **on** field so that it corresponds to the name of your joint module:



Your program should now look like this:



Next, click the play button:

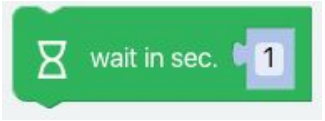
The Joint module should now move from a vertical position to 90 degrees on the X servo.

Congratulations! You have now programmed your first program!

Try changing the angle to -90 degrees and press play again. Observe what happens.

4. Move from 90 to -90 degrees

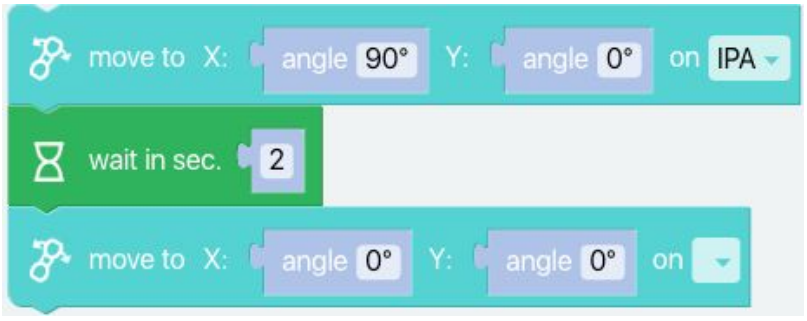
First, change the angle of X back to 90 degrees in your existing program.
You will need to insert a **wait** block. Click **Loops** and select the block called ***wait***.



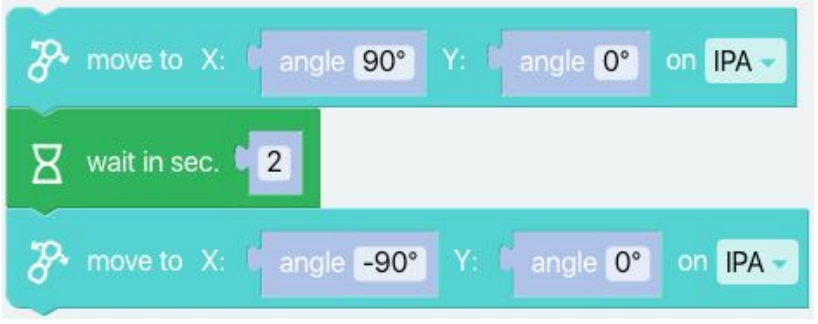
Connect this to the ***move*** block from before (drag it directly below the ***move*** block so that they click together like puzzle pieces, making a click sound).

Then change the ***wait*** block to 2 seconds.

Now, take a new ***move*** block and insert it underneath the wait block.



Change angle X on this block to -90 degrees. Remember to select the name of your Joint module.

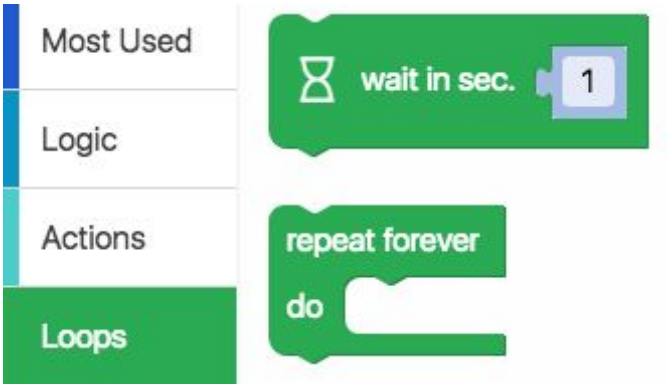


Click on run, and run the program.

Congratulations! You have now programmed your second program!

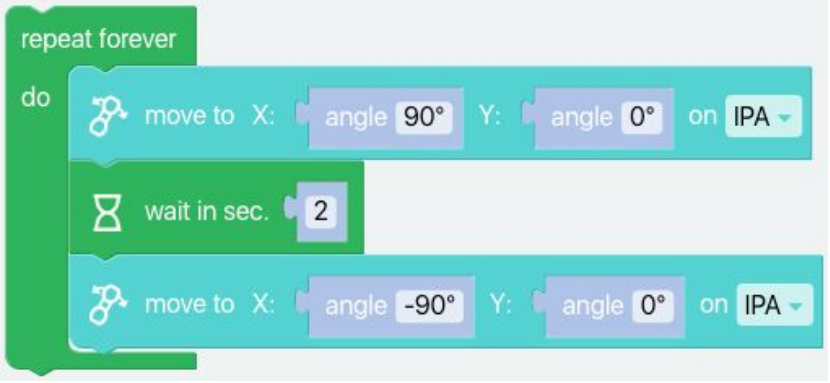
5. Introduction to loop

To make the joint module continue moving from 90 to -90 degrees, we can place a loop around it. Go back to **Loops** and select the block called ***repeat forever***.

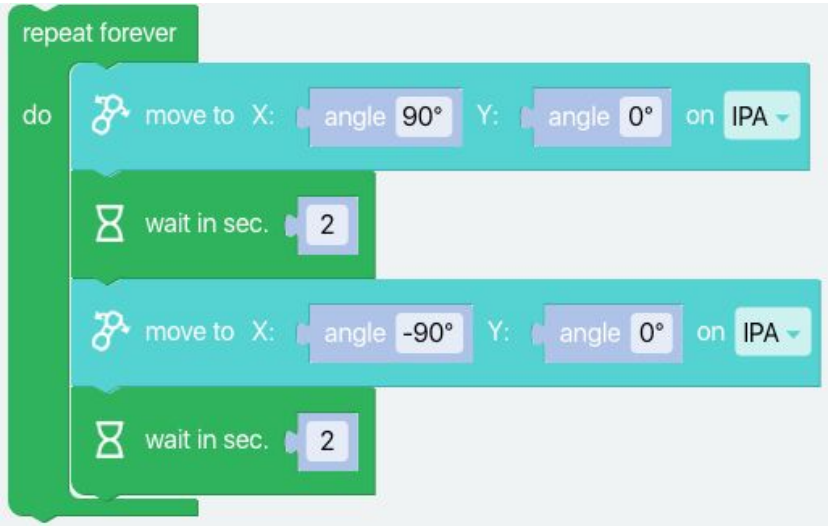


You now need to pull this loop around all the other blocks (it will expand when you connect it to the top notch, located on the first ***move*** block).

Once the loop has been pulled around the other blocks, it should look like this:



Now, the Joint module will continuously move from 90 to -90 degrees.
BUT there is a problem. When the joint module starts at 90 degrees and takes a 2-second pause before moving to -90, it works just fine, but when the program starts over, it tries to move immediately from -90 to 90 degrees. That is not possible since the joint module cannot be in two places at once.
You therefore need to insert another ***wait*** block:



Try running the program. The joint module should now continue moving from side to side until you click stop.

Congratulations! You have now programmed yet another program - this time using a loop!

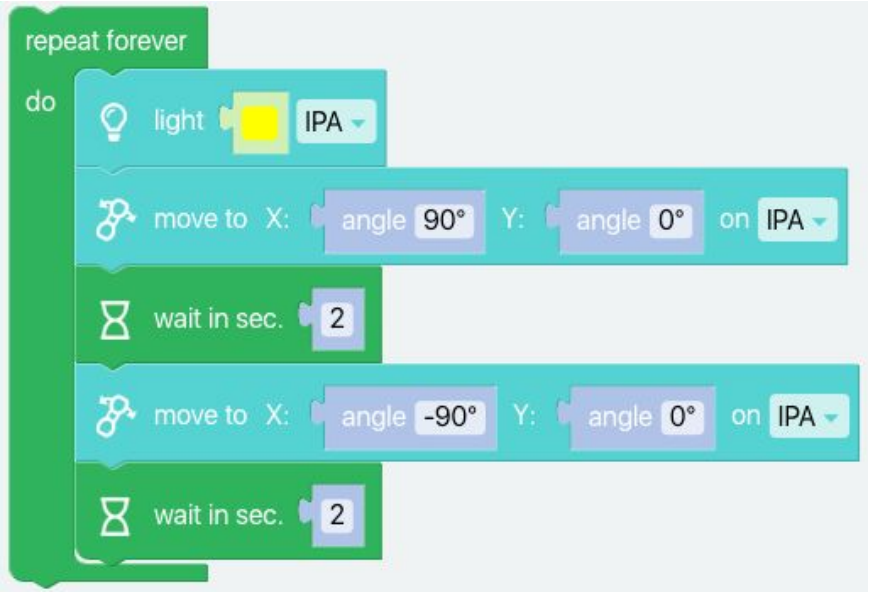
6. Program colors

You can make the Joint module blink in different colors. You can either continue working with the program you have already made and simply add the ***light*** block found in the **Actions** menu, or you can start a new program in a new loop:

The ***light*** block is inserted into our loop from before, and you can either leave the ***move*** blocks there or remove them so that only the ***light*** block remains inside the loop (remember to select the joint module's name). Note that it is the uppermost notch in the ***light*** block that needs to connect to the uppermost notch in the loop.

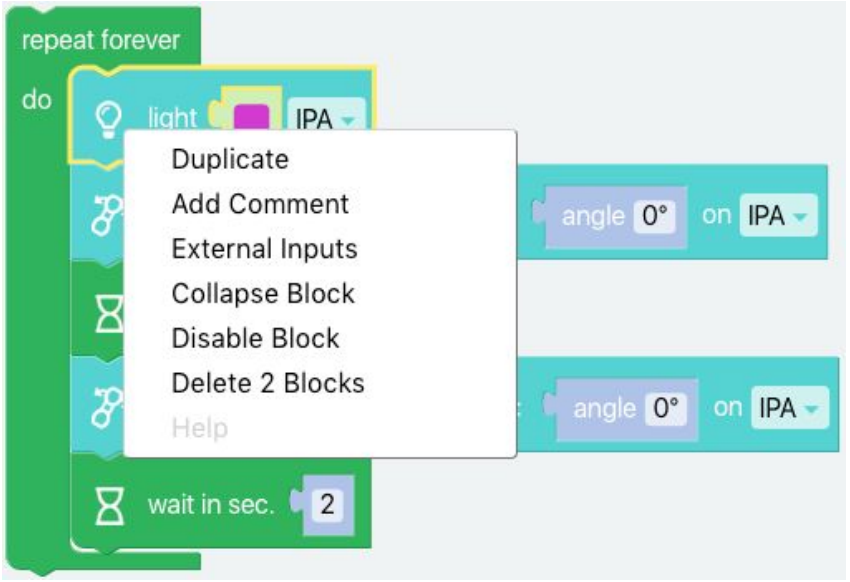
You can choose the module or the hub.

When you choose the module the program will look like this:

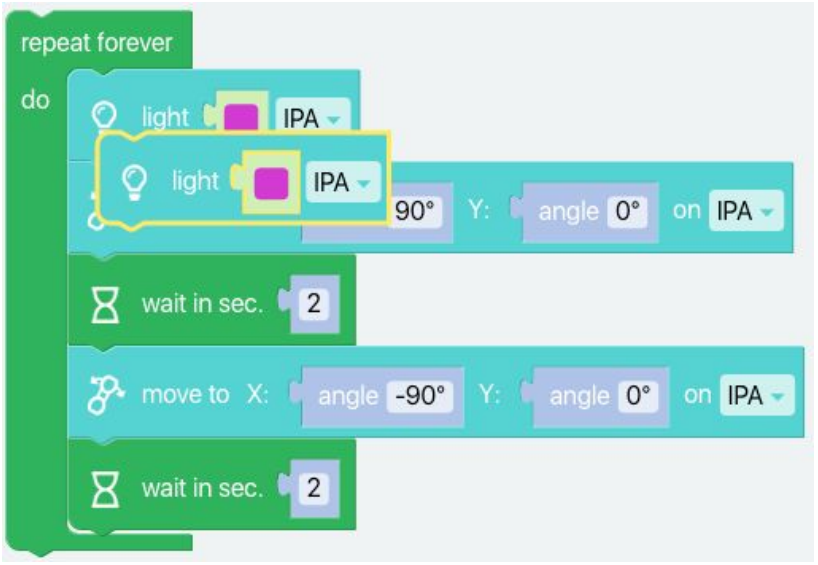


Click on the color in the block and select a color that is different from the current color of the Joint module.

Then, right-click the ***light*** block and select ***Duplicate*** or you can use CTRLl+C (you can do this with all blocks):



You should have two ***light*** blocks. All you have to do is change the color of the second block so that you have two different colors:



Drag the second ***light*** block below the ***wait*** block:

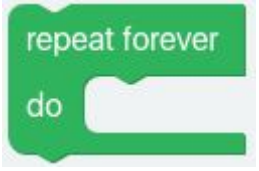
Run the program. It should now change between the two different colors you have selected. Note: that the program will keep on running until you stop it.
Congratulations! You have now programmed yet another program!

7. Learn about conditions

You can also make the Joint module perform a certain action when you press a key on the keyboard, for example. Here, the action is that the joint module moves to 90 degrees and blinks in a certain color if you press the left arrow key.

Start a new project by clicking the folder icon and choose new project. The program will ask *"Do you want to discard all x blocks?"* – you should click on the "Yes" button.

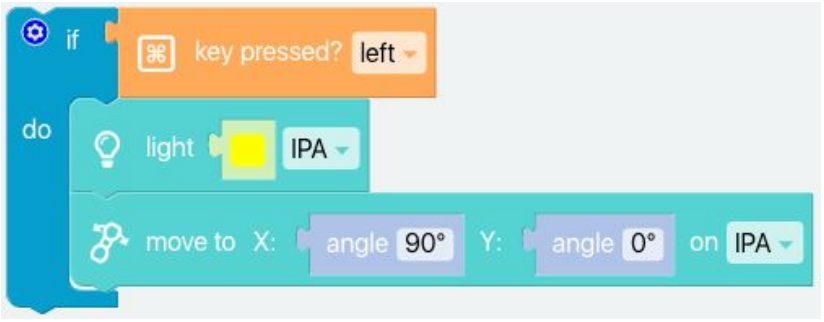
First we will start with a forever loop.



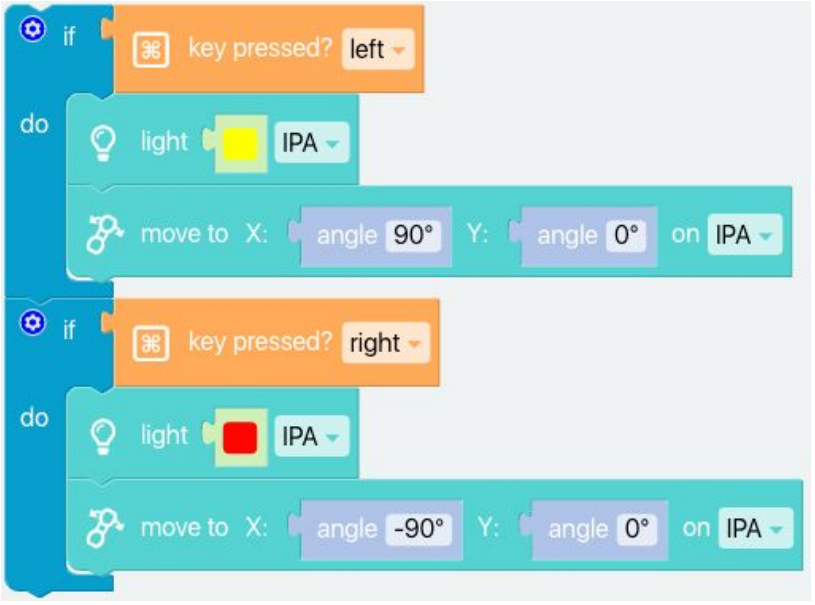
Next, select an ***if*** block from the **Logic** tab.

Then go to **Senses**, choose the block called ***key pressed*** and connect it to the ***if*** block.

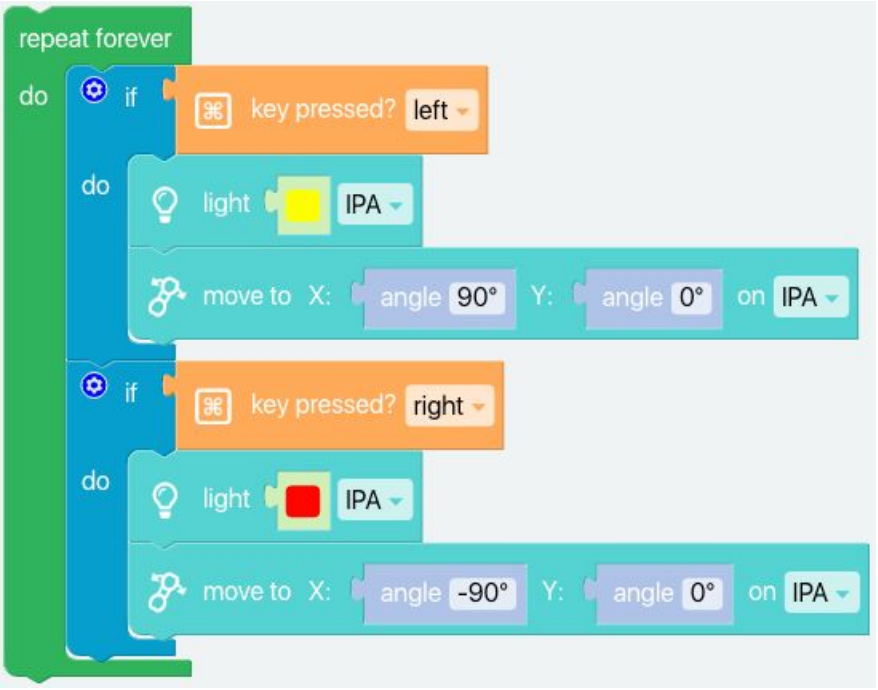
Next, edit the ***key pressed*** block - choose left key. Now take a ***light*** block and a ***move*** block and insert them into the ***if*** block - remember to insert the joint module's name and change the angle of the x servo to 90 degrees.



You can right-click on the ***if*** block and duplicate the whole thing. Now place the new ***if*** block underneath the first one and change it to ***key pressed right*** - remember to change both the color of the ***light*** block and the angle.



Now place these blocks in the ***repeat forever*** block.



Try for example including the Y servo, thereby expanding this little “remote control”.

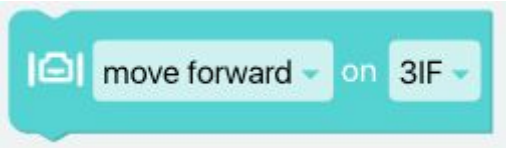
Remember that the Joint module won’t do anything until you press the arrow keys. Note that you must now actively turn off the program when you want it to stop running.

Congratulations! You have now programmed Fable to be controlled using the keyboard!

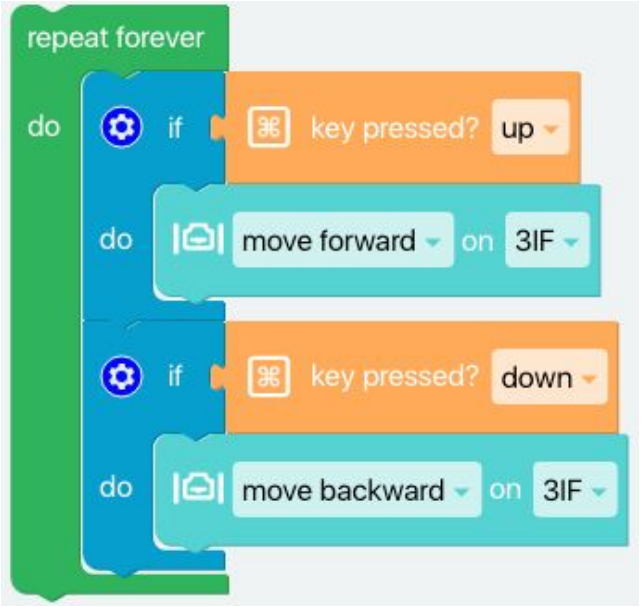
8. Control Fable Spin

The Fable Spin has two motors - A and B. They allow it to move around and carry other modules connected to it, which can also be programmed further.

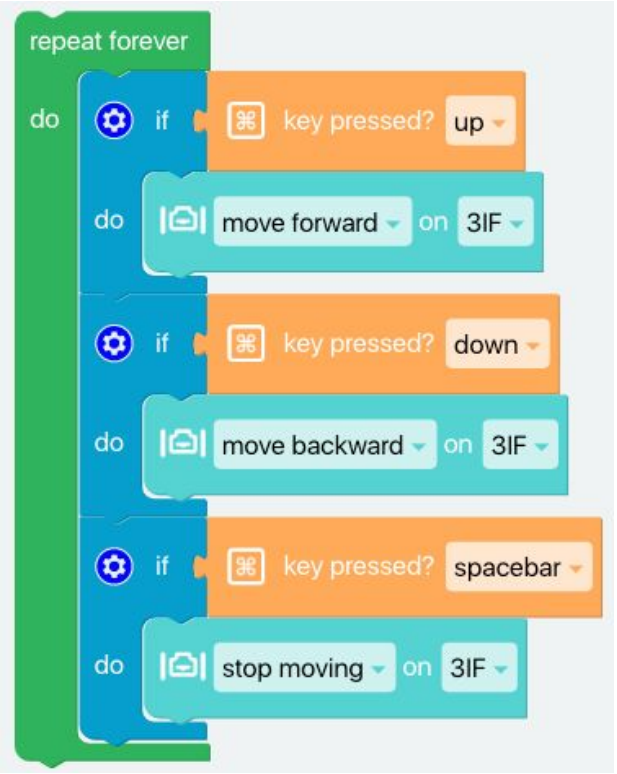
The first thing we will do is to get Spin to move forward and backwards when we use the arrow keys.
So if we take the program from before and instead of ***move*** blocks with Fable Joint we will use the Spin block ***move*** (only accessible in simple mode).



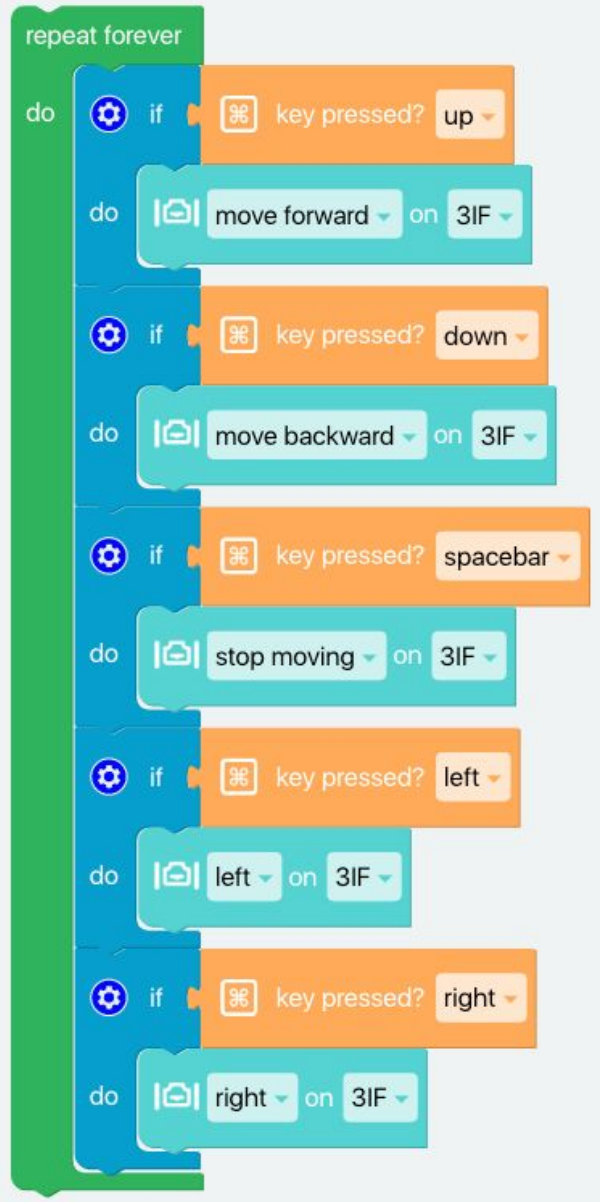
Throw the ***move*** blocks and the ***light*** blocks in the waste bin and change the arrow keys to ***up*** and ***down***. Copy the ***Spin move*** block and change it into move backwards and put it on the second ***if:***



You will also need a way to stop the robot. So what you do is copy an if block and change the arrow key into spacebar and the dropdown on the move block to ***stop moving***:



You might also want you Spin to be able to turn left and right: Copy two ***if*** blocks more and change both the arrow keys and the movement so it looks like this:



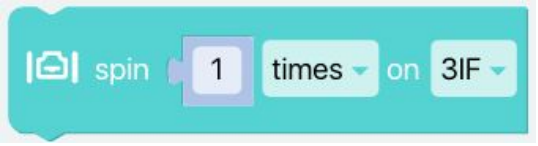
Now you have a driving robot that you can control with the arrow keys. REMEMBER TO PUT IT ON THE FLOOR BEFORE YOU DRIVE IT AROUND!

9. Fable Spin drive and turn

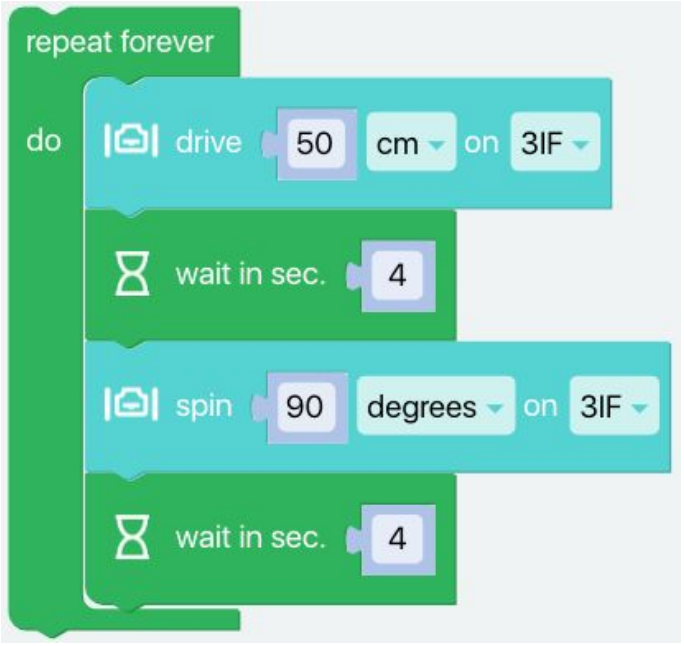
We want the Spin robot to drive in a square.
Start a new project. Take a ***repeat forever*** block. Then take the Spin ***drive*** block.

Then take a ***wait*** block and change it into 4 seconds (to make sure that it actually reaches the 50 cm).

To make the robot turn 90 degrees, take the block ***Spin***.



Change it into degrees and write 90. Put it into the loop and add another ***wait*** block in the end.

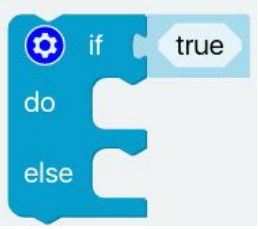


Start the program and watch your Spin drive in a square!

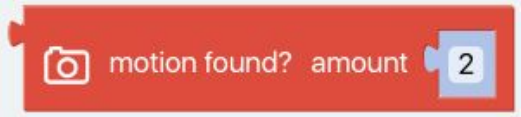
10. Make a surveillance robot

Fable can also be a surveillance robot.
Now you want to make the Joint module react when there is motion on the computer's camera or a webcam.

Again, start a new project.
Once again, you need a ***repeat forever*** block and an ***if*** block - but this time we will take the block called ***if - else***:

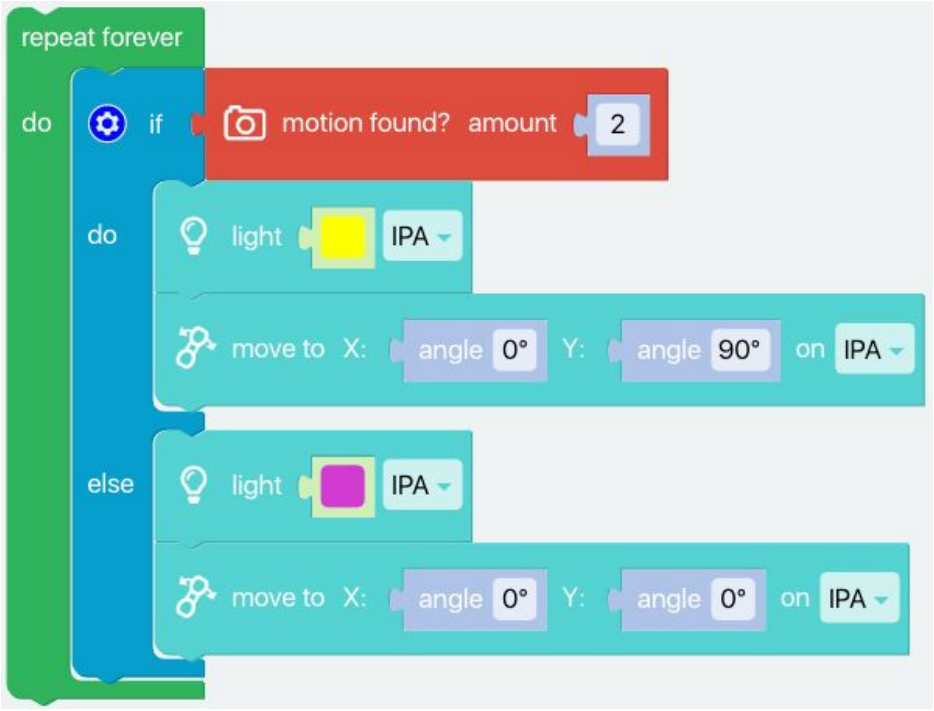


Next, under **Camera**, select the block called ***motion found?***



Connect this block to the ***if - else*** block where it says true.

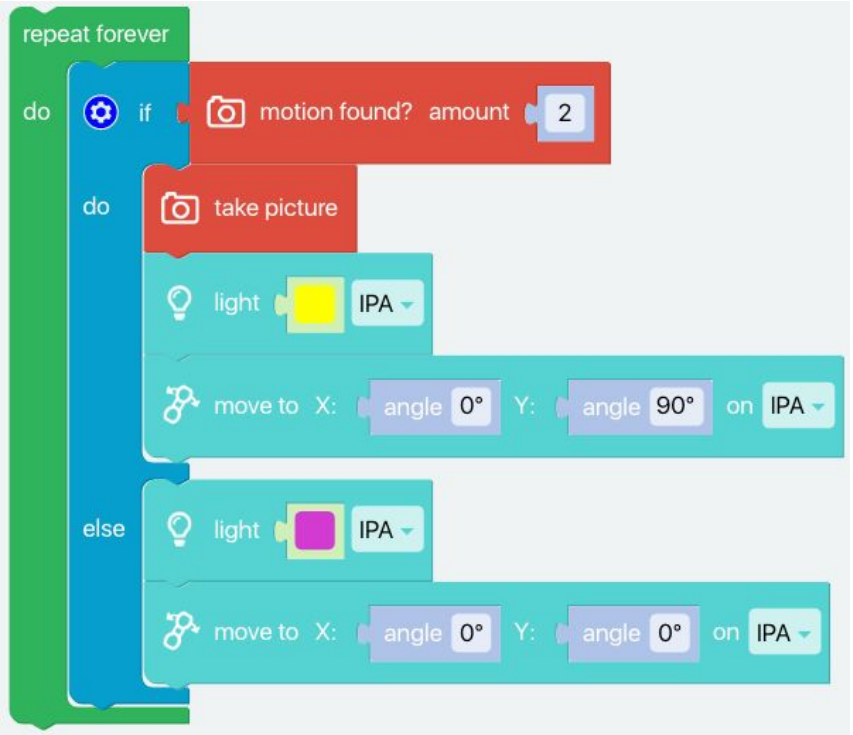
Now, you need to insert some blocks that make something happen when the camera detects motion. In this little program, the action inserted into the do block makes the joint move to 90 degrees on the Y-axis. Under ***else***, the Joint module has been set to return to 0 degrees.

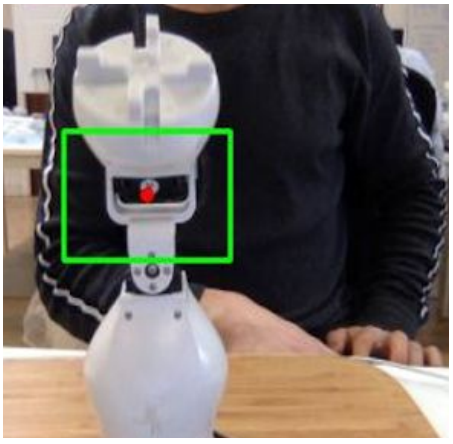


Try running the program.

Congratulations! You have now programmed yet another program!

You can also make the program take a picture when it detects motion. The program will look like this:





Note: that the camera will open in a pop-up window.

The pictures that the program takes are saved in a file in the documents folder under Fable called ***"Fable pictures."***

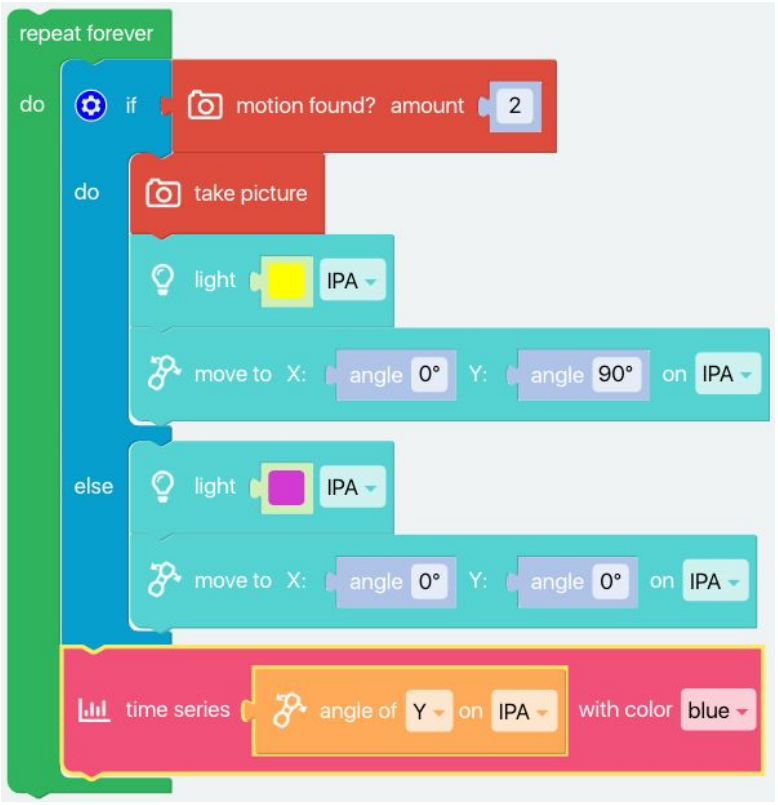
11. Make a graph

If you want Fable to make a graph over the movement of the Joint module (or it could also be from a sensor in Fable Spin or from a sensor in the phone). Start by inserting the block called ***time series***, found in the ***Data*** menu:

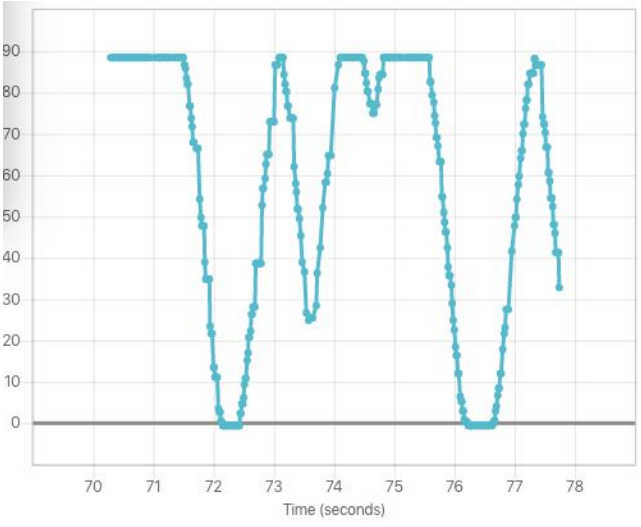
Next select the block called ***angle of*** from the ***Senses*** menu:

Insert the ***angle of*** block into the ***time series*** block (where it says “5”) - change to the Y-motor:

Finally, insert this new block into the program you built before. Below, the angle has been set for the Y servo:

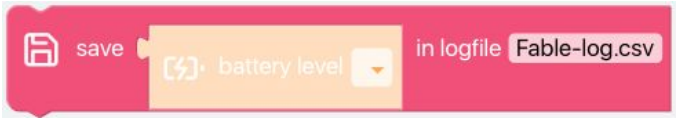


Here is an example of the graph:

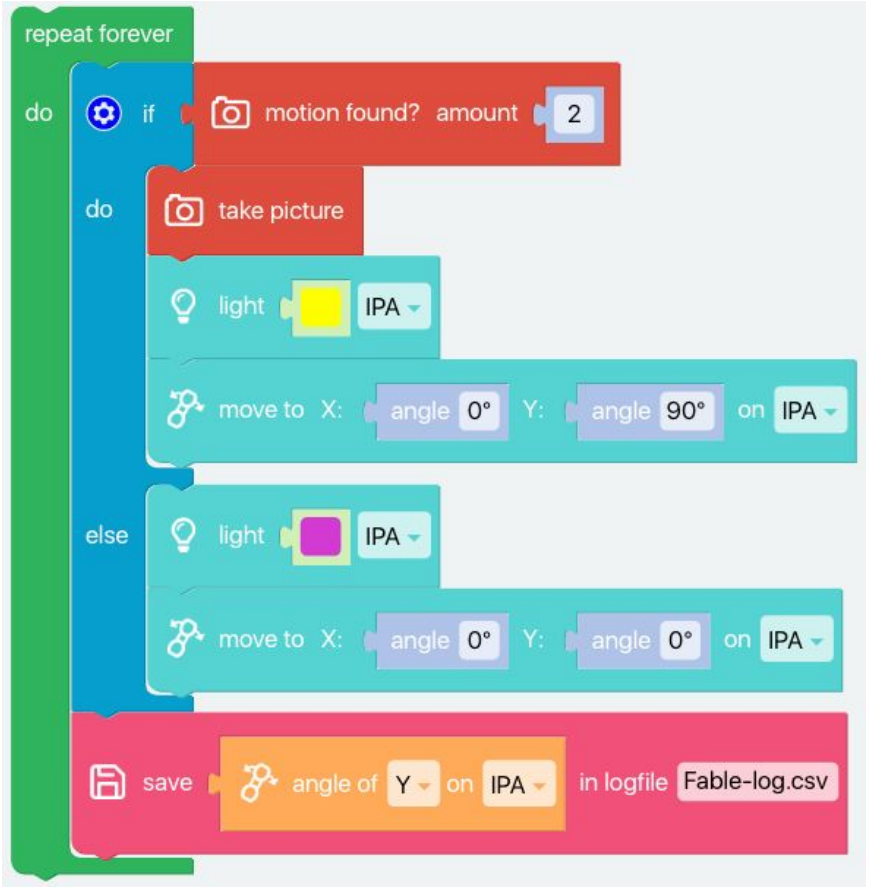


12. Log data with Fable

You might also want to save the Joint module's movements in a log file.
You do this under ***Data*** by selecting the block called ***save in logfile***:



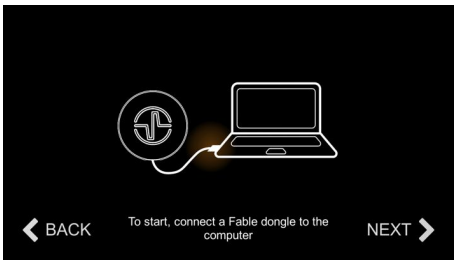
Just as you did before, insert the ***angle of*** block into the ***save in logfile*** block. If you add to your program from before, it will look like this:

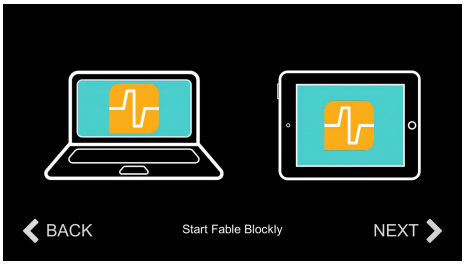


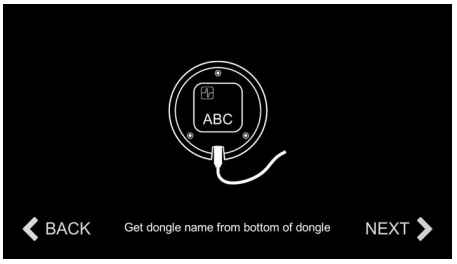
When the program is running, it will create a file in *Documents/Fable* called "*Fable-log.csv*" with a log over the Joint module's movements in degrees.

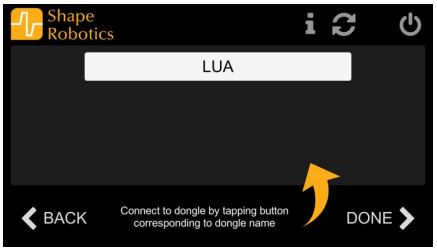
13. Fable Face

There is an app for smartphones (iOS and Android). It's called "Fable Face" and can be downloaded via the App Store or Google Play.
Once you open the app you will be guided through these steps:

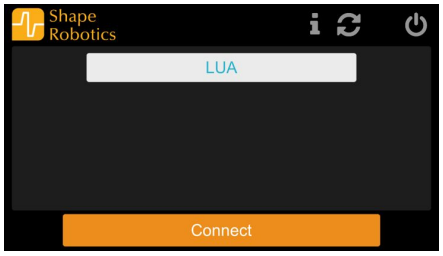


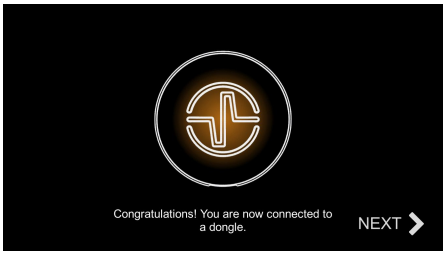


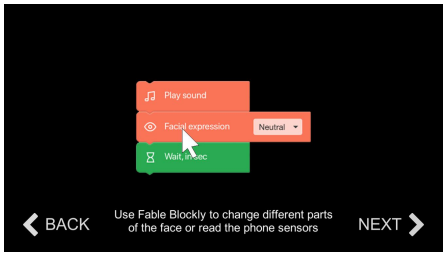


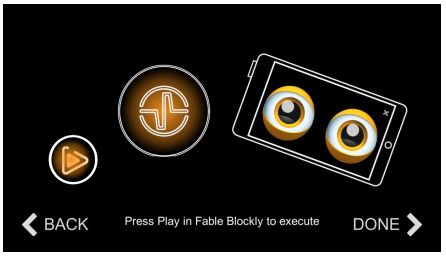


When you click on the button corresponding to the Hub it will change color and you can click connect.

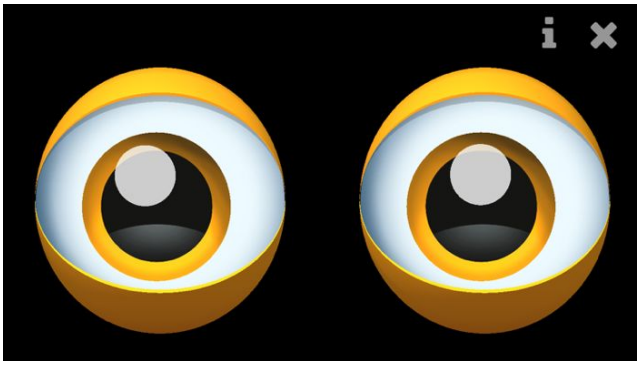






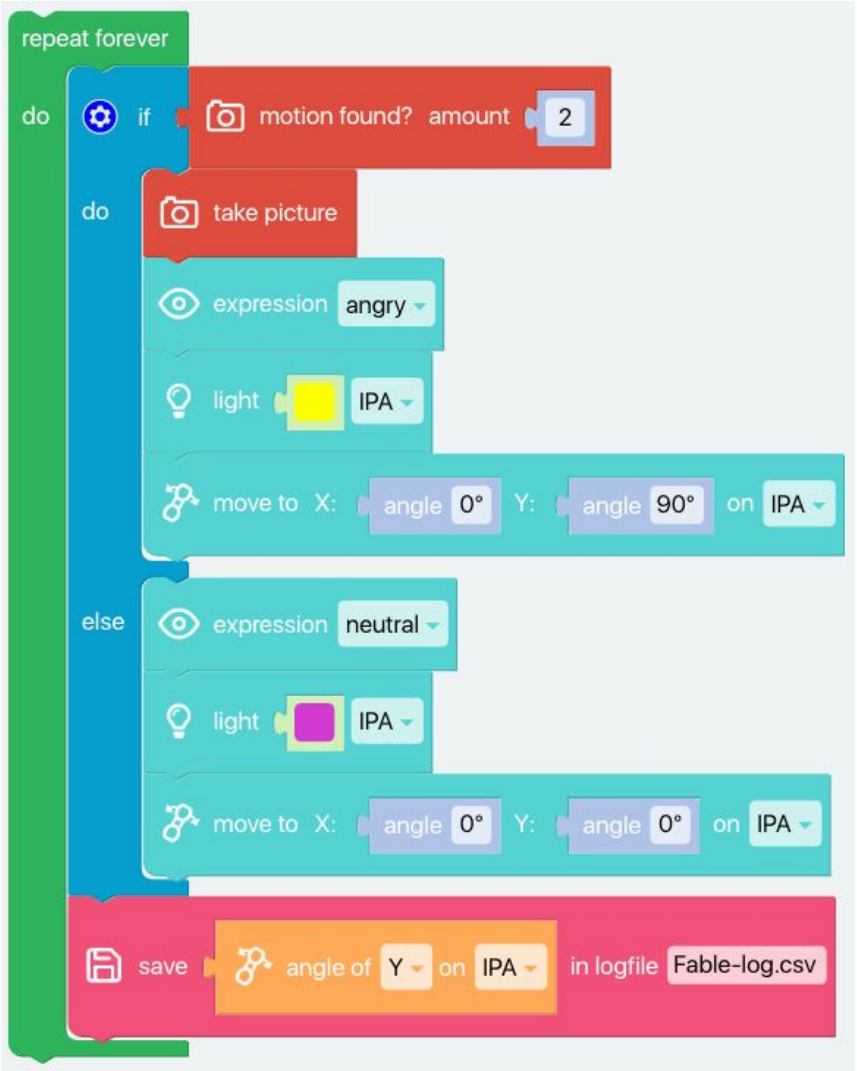


Afterwards the screen will change into the Fable Face's eyes.



Under ***Actions***, select the block called ***expression***.

If you're continuing with the program from before, this block should be inserted in two different places. One into the first ***if***, where you change the facial expression to **angry**, and one into the ***else*** part, where it should just be **neutral**.

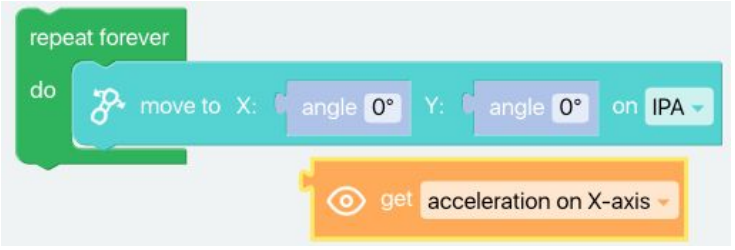


When you run the program, the facial expression will change when there is motion.
If you have the phone holder that comes with the Fable set, you can use it to attach the phone to the Joint module otherwise, you can build a holder out of LEGO and use that to attach the phone.

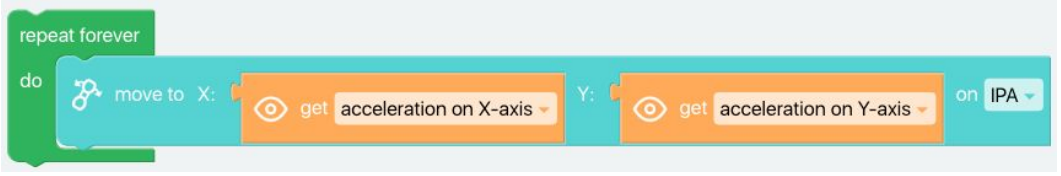
It can also be a good idea to insert a ***wait*** block before the ***else*** part, so that the joint module has time to finish before it has to change its facial expressions. You might even want to insert a three-second pause instead of just one.

14. Control Fable with smartphone

To get the app to control the Joint module, you need the ***get acceleration*** block from senses inserted into a ***move to*** block. This time, start with a new program, using a ***repeat forever*** block:



Next, duplicate the ***acceleration*** block and drag it inside the ***move to*** block, replacing the grey **angle** blocks - both X and Y (the blue blocks are thrown out). Quindi, modifica uno dei blocchi di accelerazione sull'asse Y.



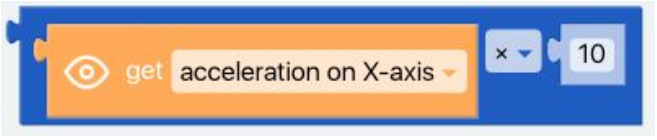
Test your program. The joint module should now (more or less) follow the movements of your smartphone.

Congratulations! You have now programmed a program that controls a robot via a smartphone's accelerometer!

16. Refine the program

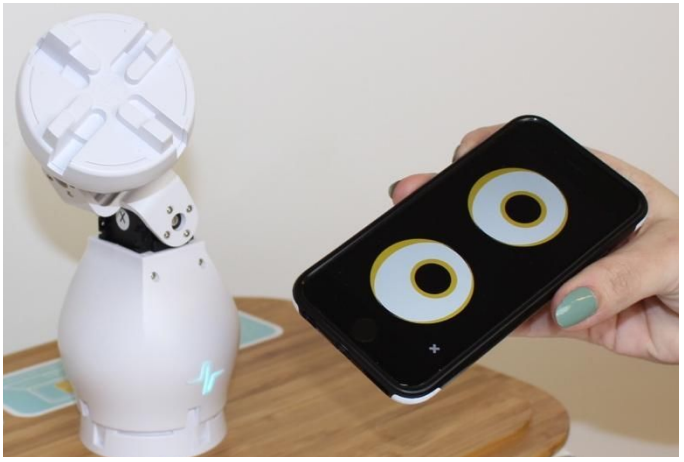
Perhaps you have noticed that the Joint module does not move very much when you move the smartphone? You can fix this by multiplying the movement of the smartphone by a certain factor, e.g. 10. First, click ***Math*** and select this block:

Next, drag one of the ***get acceleration*** block used in the previous exercise and insert it into the beginning of the math block and then change the number to e.g. 10. It will now look like this:



You will have to do the same with the Y-axis (or you can just duplicate it and change it to Y).

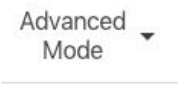
Run the program again and see whether the Joint module moves a bit more.



This program can be used for many different things. For example, you can use it for the maze game, which is one of the accessories for Fable. Find the activity at: shaperobotics.com/activities.

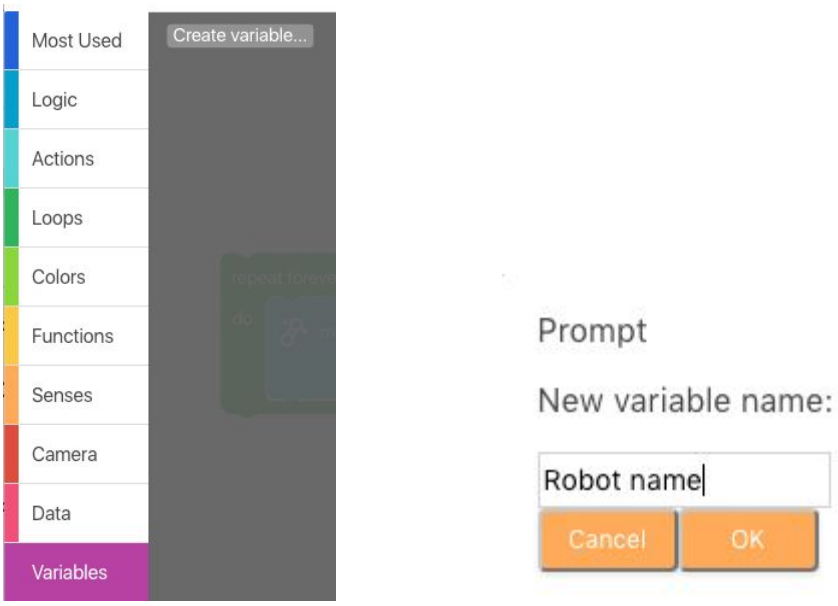
17. Make a variable

If you want to create a longer program, it might make sense to use a variable.
Instead of having to change the ID of the modules on every single block, you only have to change it in one place.
But first we need to change Blockly into ***advanced mode***. Click in the bottom left corner (where it says advanced).

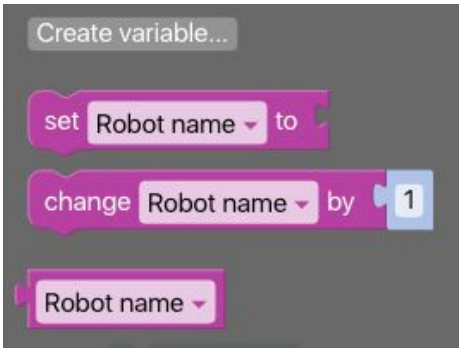


Now then menu will change and there are 3 new menu points: ***functions***, ***variables*** and ***lists.***

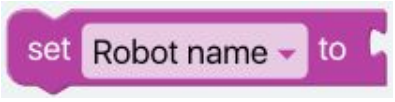
Now click ***Variables*** and create a new variable, for example called “robot name”.



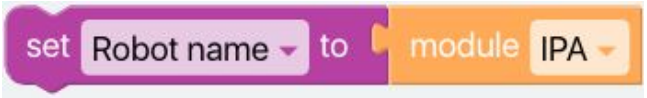
You will then see a number of new blocks (the ones you have just created).



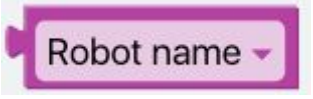
Select the one called ***set robot name to***:.



Next, go to the ***Senses*** menu and select the block called ***module*** and connect it to the other block (remember to select your module).

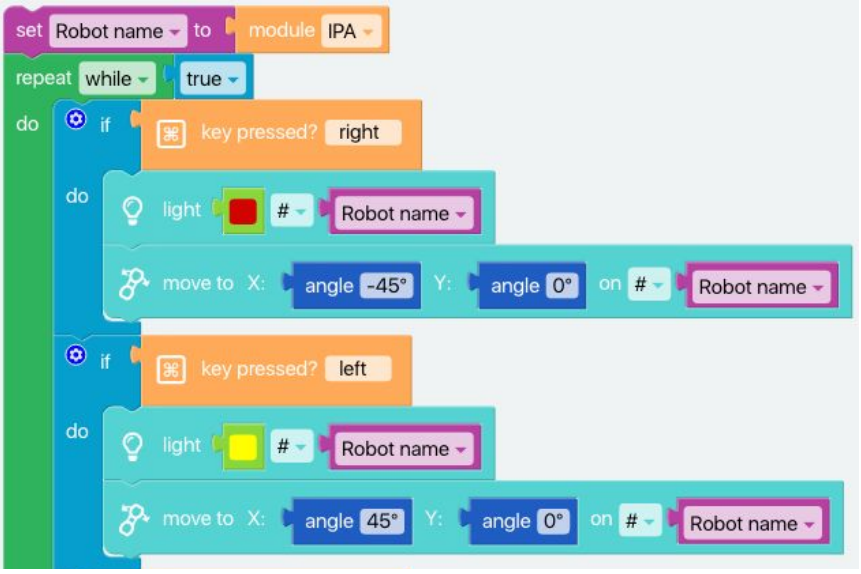


We can then use the block called **robot name** and insert that instead of the module’s name.



If you choose the # icon instead of the module name, you will get an extra field into which you can insert the **robot name** block.

If you have a remote control program like the one from exercise 7, it will look like this:

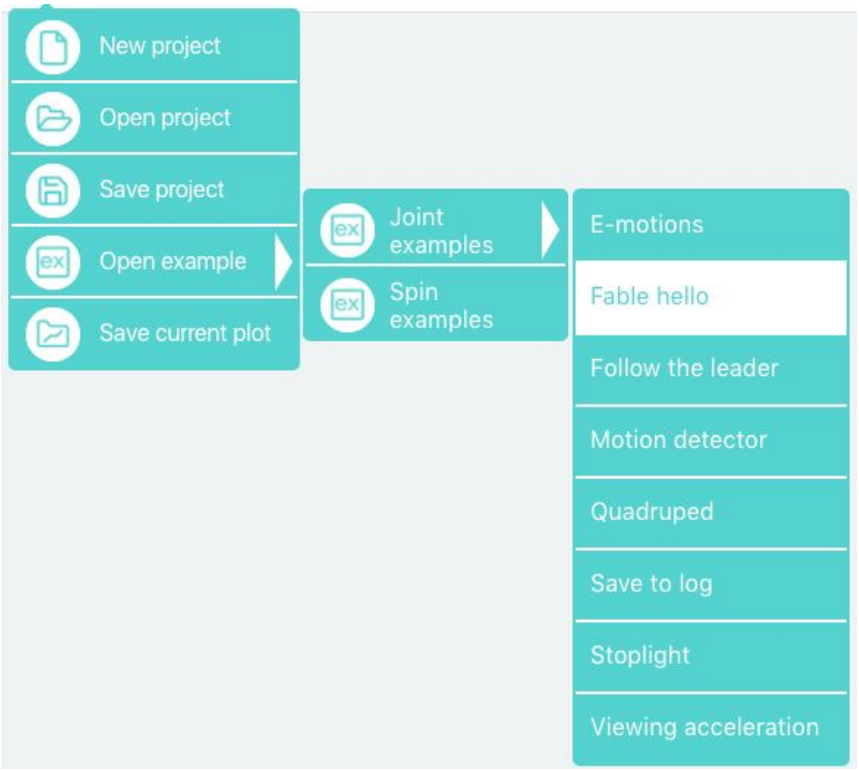


It would now be easy to switch the Joint module (for example if it runs out of battery). Here, you would only have to change the name in the ***module*** block.

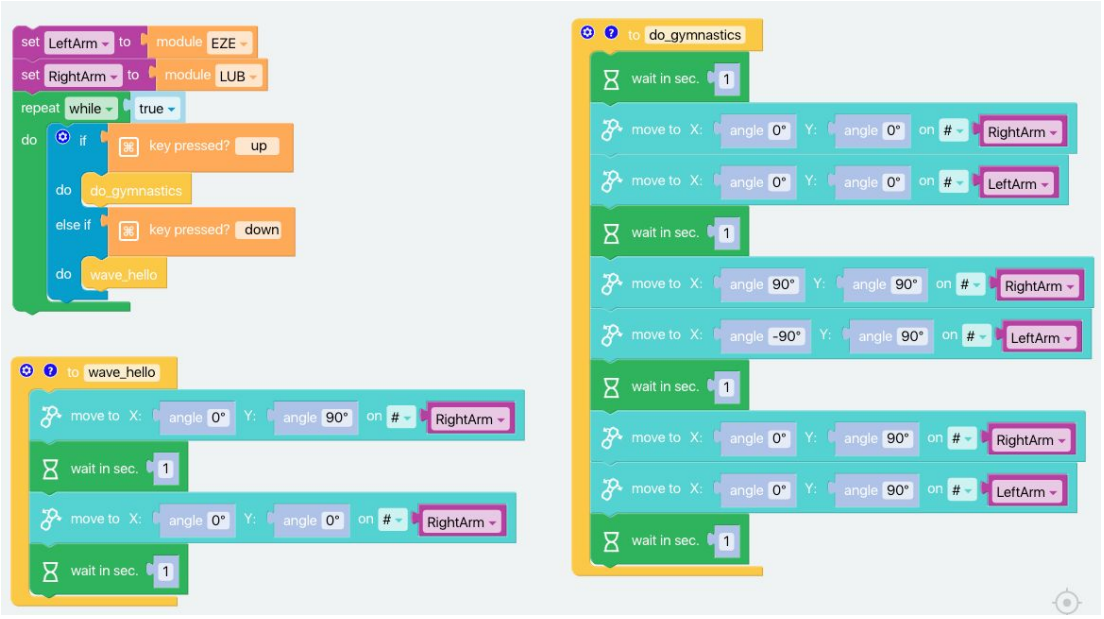
18. Save, open and examples

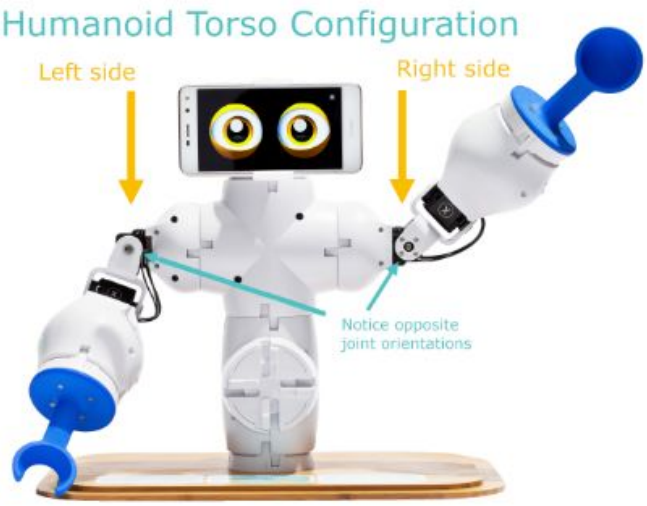
In Fable Blockly you can save a file like in other programs. Click on the folder icon:

Here you can start a new project, open a project or save the project that you are working on. You can also go into the ***open example***.



E.g. take the ***Fable Hello*** and you will see that it opens a program that is already made and you can see a picture of how to build the robot.





Congratulations! You are well on the way to be a Fable-programmer!

The Fable system is a great tool to e.g. use in a classroom.
The students could work as engineers, trying to solve real world problems. If your approach as a teacher is in a playful learning way, the students could learn to be very creative and innovative.

Find inspiration on our website and on our youtube channel.
We have held a ton of workshops both with teachers and pupils where we have used that kind of approach. See pictures and videos on our website or our youtube channel.
You can find more Fable activities here: shaperobotics.com/activities
Our structured lesson plans are available here: shaperobotics.com/portfolio/lessons/

Happy programming!

Shape
Robotics

Rugmarken 18, 3520 Farum, Denmark
shaperobotics.com