Inhalt

Preface

Mindstorms Robot Inventor Set 51515 continues the successful history of the Mindstorms series. The new generation is compatible with the electronic components of Boost 17101, Spike Prime and Powered-Up.

As the decisive difference to Boost and Powered-Up, Mindstorms is not generally “remote controlled” by a PC or Handeheld but the programs can run independently on the Hub. Unfortunately, the official set only comes with models that are remote controlled. Whereas the Bob-It-Inventor can be used independently from a PC, as soon as the program is uploaded to the hub.

The Document has three segments:

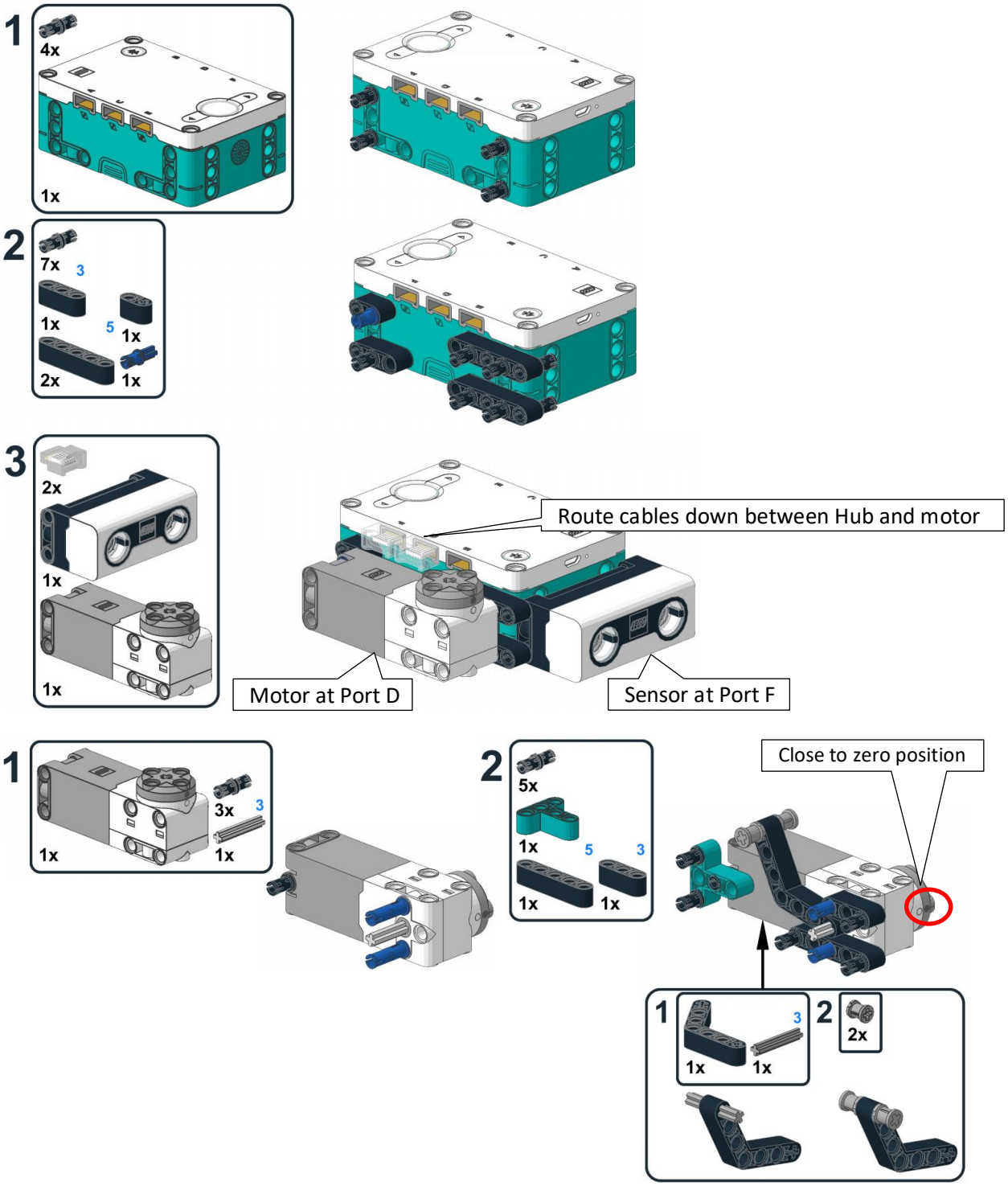
- BUILD – Building instructions
- CODE – Programming the Hub
- Play - Operate and customize

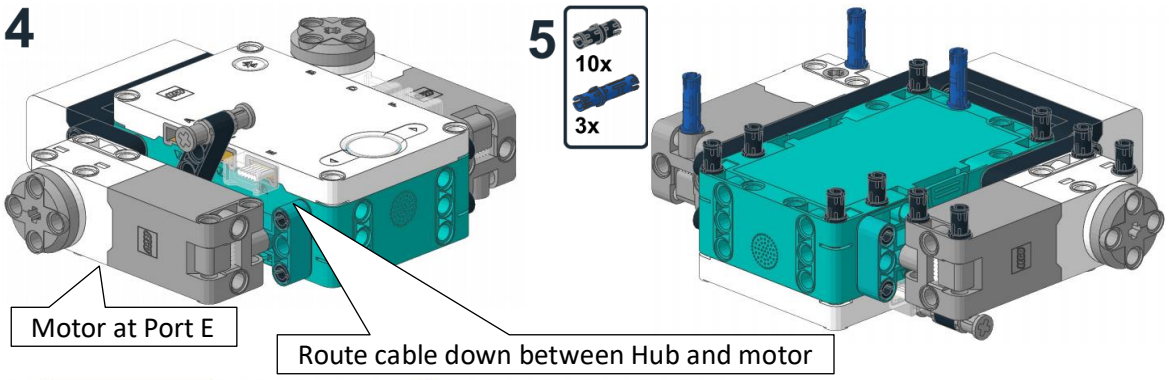
Bob-It-Inventor – the fast reaction game

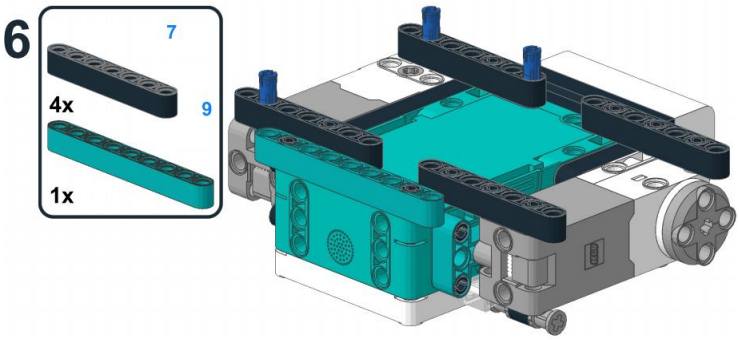
The Bob-It-Inventor is a reproduction of the Bob-It game by Hasbro.

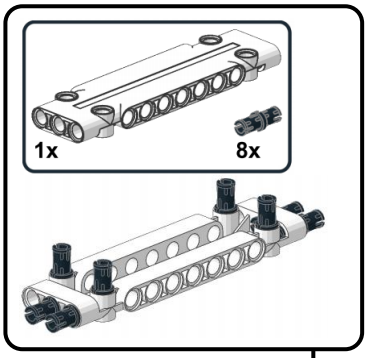
Building Instructions

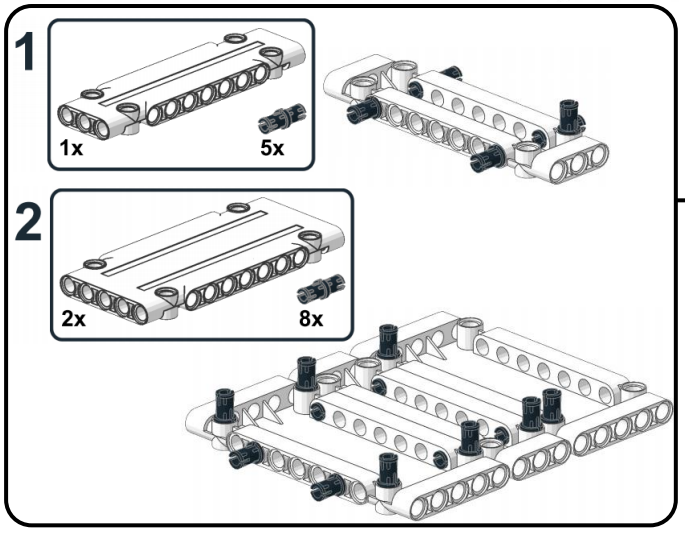
Bob-It-Inventor can be built with just the parts in the 51515 Mindstorms Inventor set. You do not need any additional parts besides a PC, tablet, or phone to upload the program.

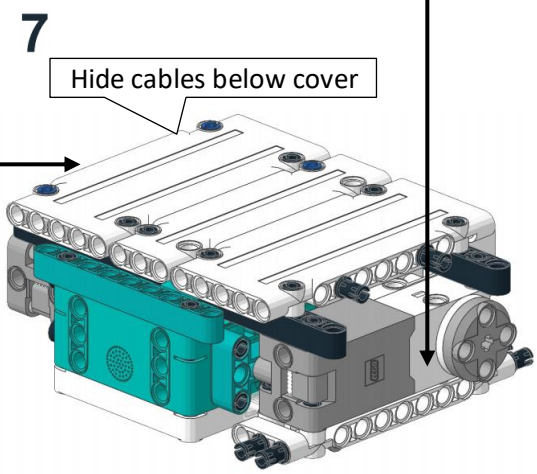


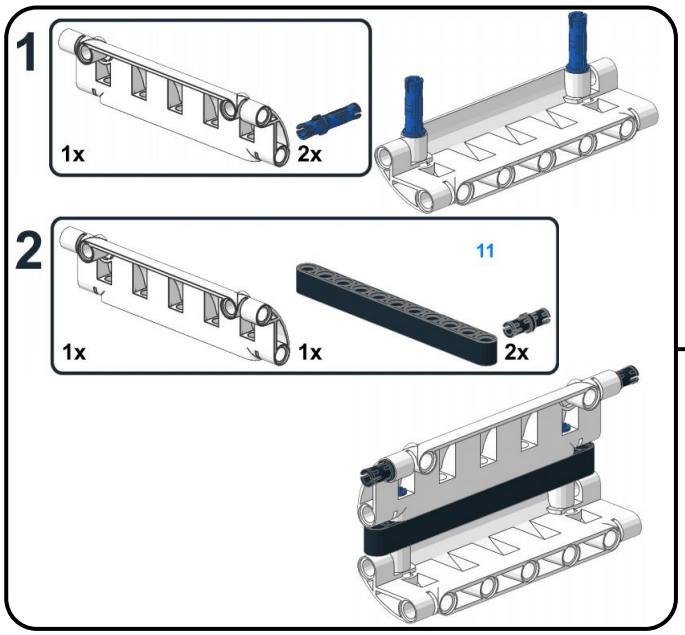

4
5
10x
3x
Motor at Port E
Route cable down between Hub and motor

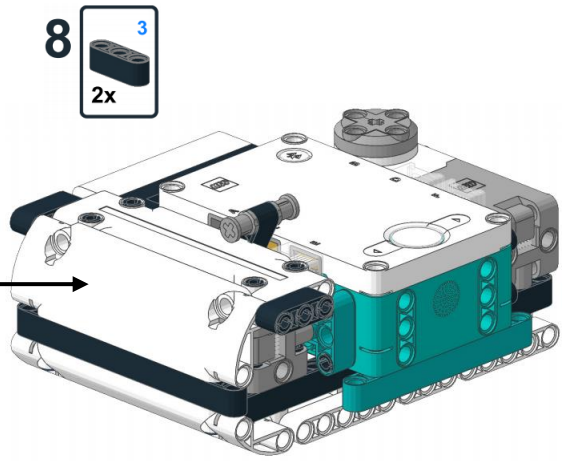

6
7
4x
9
1x

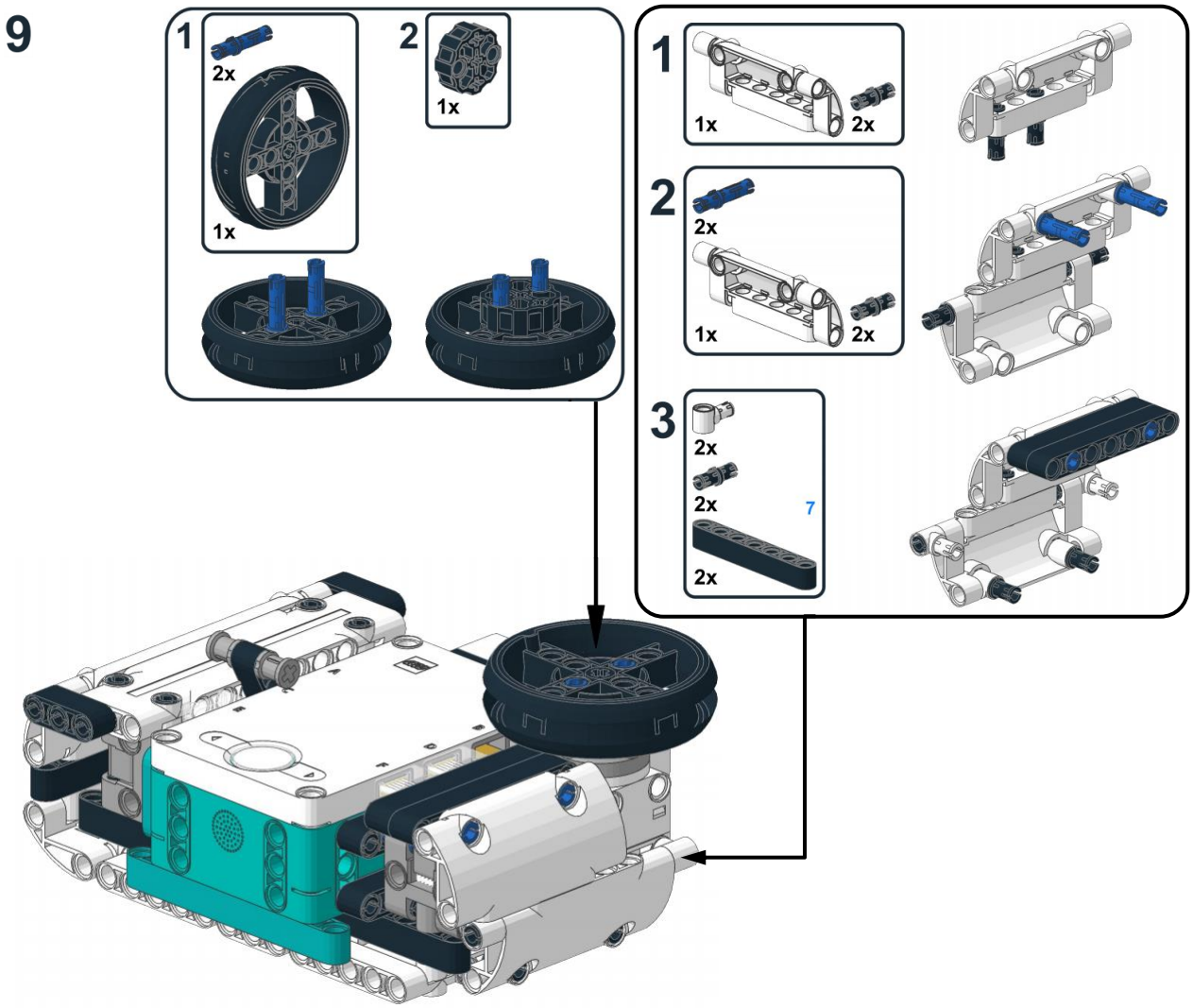

1x
8x


1
1x
5x
2
2x
8x


7
Hide cables below cover


1
1x
2x
2
11
1x
1x
2x


8
3
2x



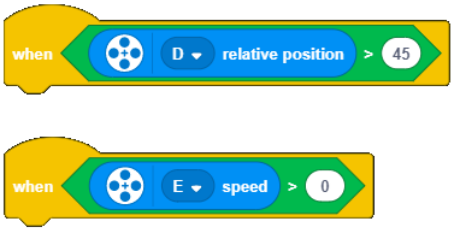
For trouble-free gaming, it is important to properly hide the cables between the lower cover and the hub. The cables can be routed from the plug directly down to hide them below the hub.

CODE – Programming

Use motors as movement sensors

Besides using a motor to provoke movement, a motor can also be used as a sensor for movements.

Each motor report on its speed, absolute position, relative position, power, and an interruption of a planned movement. When used as a pure sensor you will most often evaluate speed or position. The speed or position can be compared to a threshold value to start a program sequence via a `when…` block or take an `if…then` decision.



Variables

Variables work as the memory cells for a program. Contrary to script-based programming languages, you do not have to worry about the correct definition of each variable in the WordBlock language within the Mindstorms App.

The BobIt-Inventor program remembers the state `GameOver` as well as the correct execution of a task in `completed` by assigning a `0` or `1` to these variables. In script-based languages you would use a Boolean variable to just remember `False` (`0`) or `True` (`1`).

The randomly generated number of the next task is stored in variable `task`. It is used to check any player move against the requested task.

A variable can also take the result of an equation, even if this equation involves the variable itself. The `time` limit for a move is multiplied by `0.9` to reduce it by `10%` and the result is directly assigned to the same variable `time`, again.

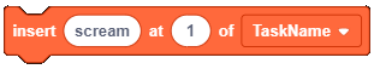
The `change…by` block is used for simple counting. In our case, we use it to keep track of the players score by counting the successful moves. The `change…by` block can also be used to count backwards `-1` or change the variable by any other value.

The value of a variable can be used in calculations, as motor parameters, be displayed on the light matrix or be used in `if…then` statements to take decisions. The BobIt-Inventor uses variables for most of these examples.

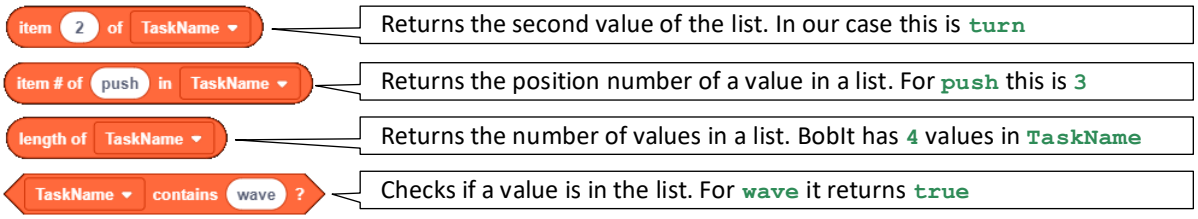
Lists

In addition to simple variables, a WordBlock program can also handle lists. A list can store several values in an organized sequence. These values can then be retrieved one after another or can be addressed called individually.

Similar to variables, lists are capable to take values even before you start the program, e.g. via the monitoring menu on the right side of the screen. Variables and Lists keep remembering their values even after a program has ended. Therefore, the content of list `TaskName` is deleted at the very start of the program and the new values are assigned one after the other. BobIt-Inventor uses the list to store the names of tasks as strings to later use it to broadcast the required move. A list can be fed with values in several different ways. Values can be added one after the other, as shown above. You can insert a value at a specific position. Or you replace the value at a specified position with a new value.



When working with lists, you can retrieve a value from a specific position of a list, search for the position of a value, get the number of values in a list or check if a certain value is within a list.



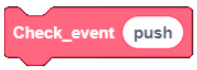
Broadcast messages

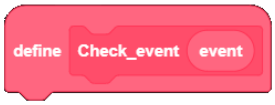
Broadcast messaging can be used to start several program sequences at a time. BobIt-Inventor therefore only uses the broadcasting blocks which do not wait for a sequence to be executed before the program continues its execution. This way, you can play some tones while changing the animation on the light matrix or resetting the clock at the same time.

BobIt-Inventor also uses broadcasting blocks to elegantly branch off into different program parts based on the position of move names in the list `TaskName`. The code branches off into the display of randomly selected moves, without stringing together several `if…then` blocks.

My Blocks

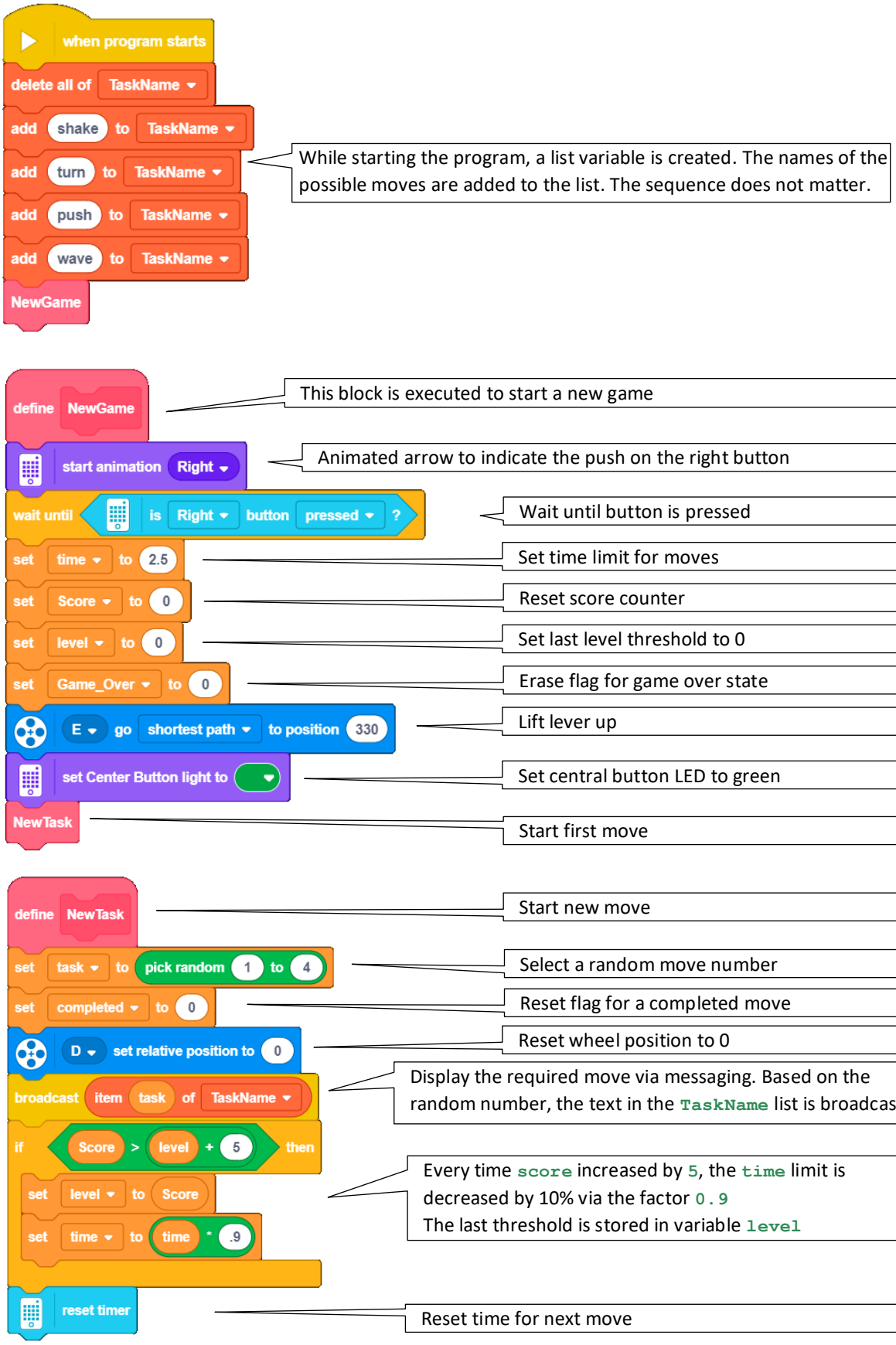
Similar to broadcast messages, block sequences can be grouped under a self-defined “My Blocks“. In contrast to broadcasting block, the call of a “My Block” stops the execution of any block below the “My Block” till the entire “My Block” sequence is executed. While calling a “My Block”, one or several parameters can be submitted to the “My Block” sequence. BobIt-Inventor uses this functionality to transmit the executed move to the Block `Check_Event`.

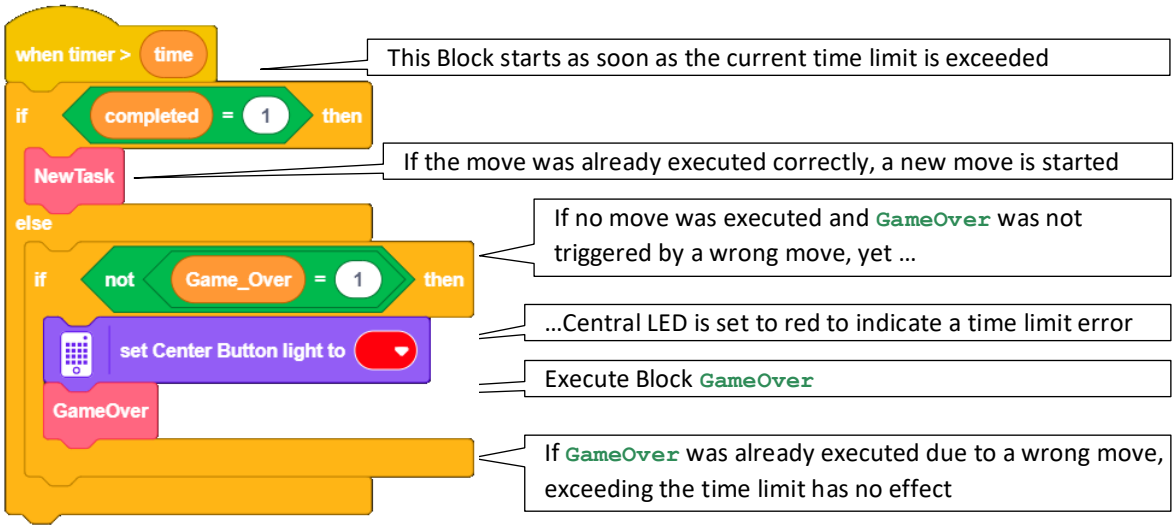


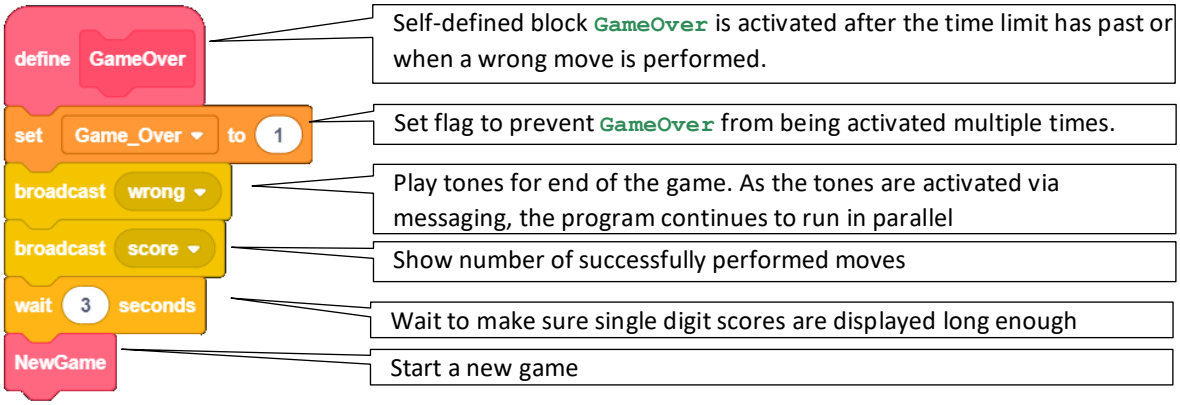


The WordBlock program

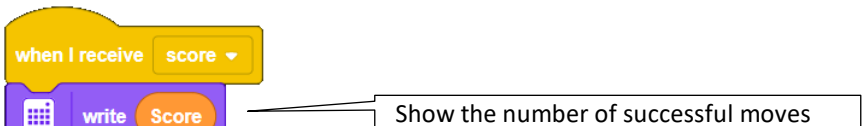
Bob-It-Inventor is programmed via WorkBlocks. The code is divided in several sub-programs which are all placed on the same programming canvas.

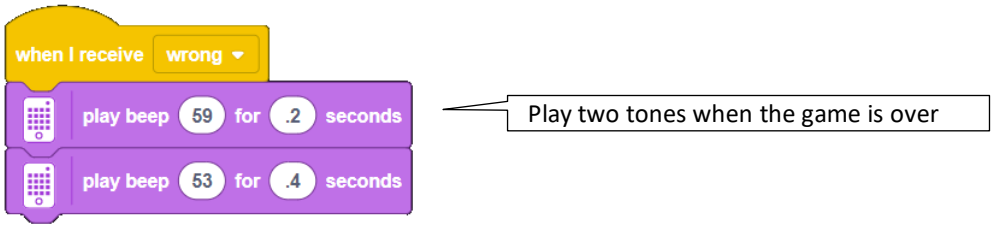


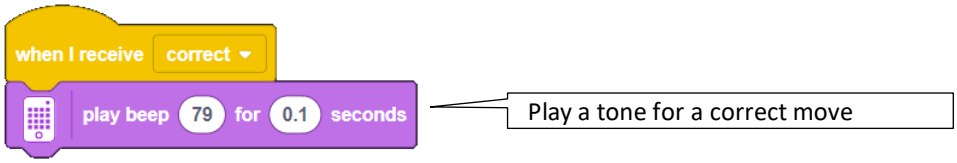


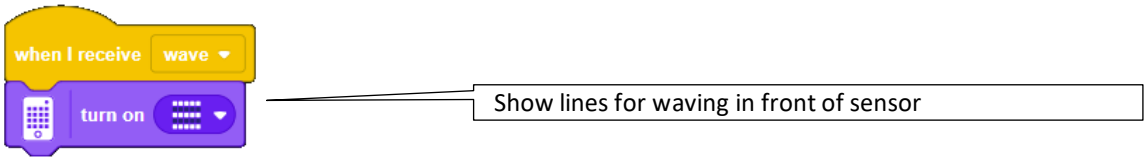


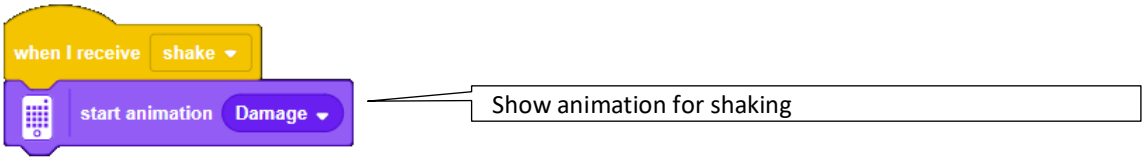
The following blocks are used to receive broadcast messages. The messaging enables us to let several parts of the program run in parallel. Furthermore, we can elegantly realize to branch off to several different program parts based on a variable.

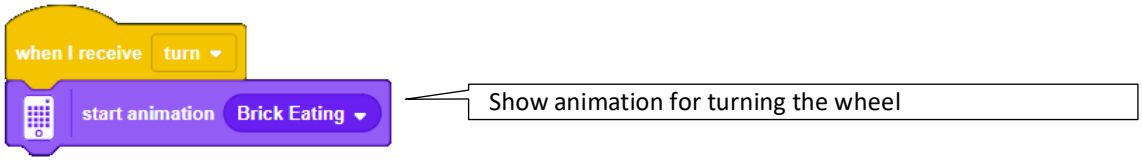


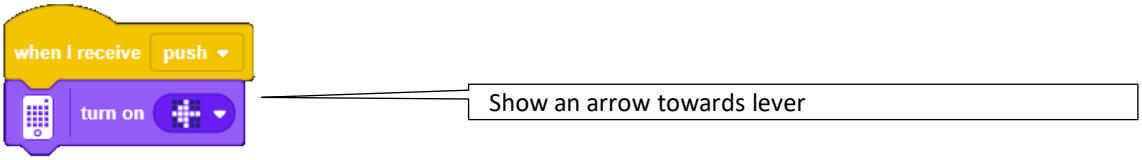






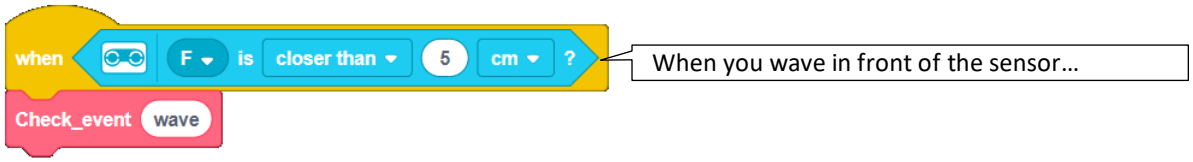


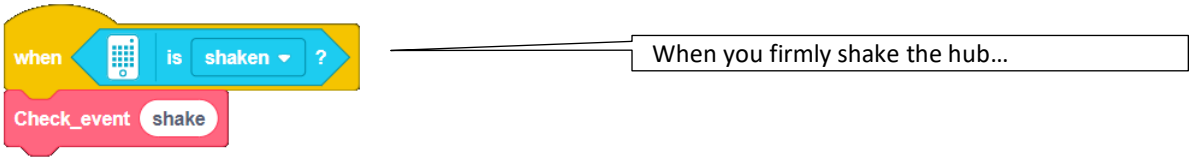


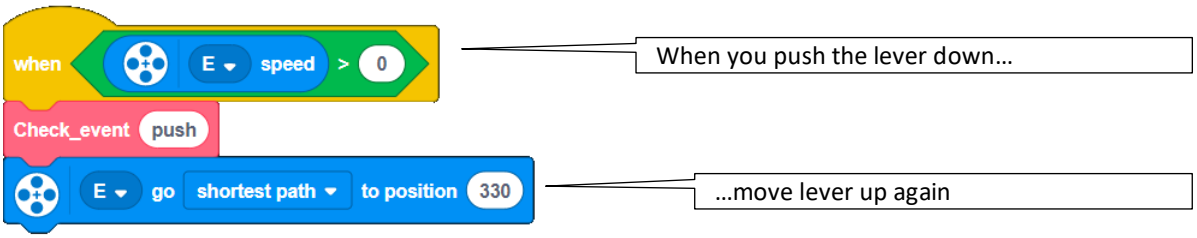


You may want to create your own animations or symbols to display the different moves.

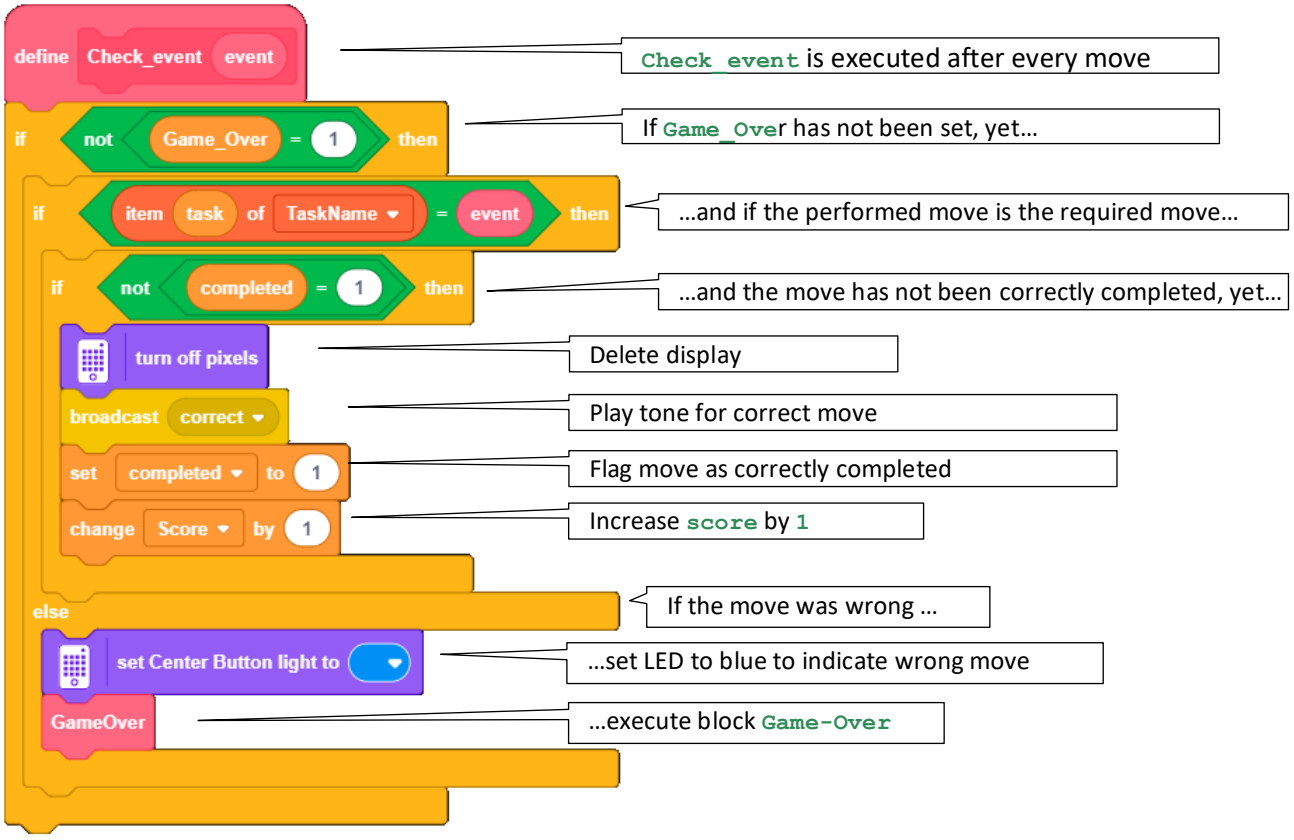
The following blocks are executed as soon one of the monitored events occurs. Every event triggers the same block `Check_event`. When the block `Check_event` is called, it receives a text as a parameter to describe the event.







The self-defined block `Check_event` and its parameter `event` checks with each execution if the performed move matches the required move. It then triggers an adequate reaction.



Install downloadable program on the Hub

As an alternative to creating the program yourself, you can download a ready to use version and upload it via the LEGO Mindstorms APP.

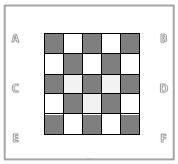
`Bob-it.lms`

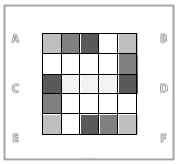
Open your Mindstorms-APP on a PC. Via the menu „File“ and „Open...“ you can load any `.lms` file into the APP. The program can then be uploaded to the hub the same way you upload any other program to the hub.

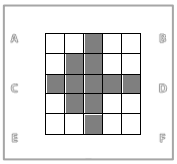
Play

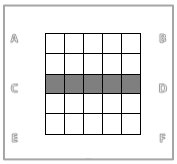
When starting the program, the display shows an animated arrow to the right. Pushing the right arrow button starts a game.

Different moves will be displayed at random. The player must perform these moves within the given time limit.

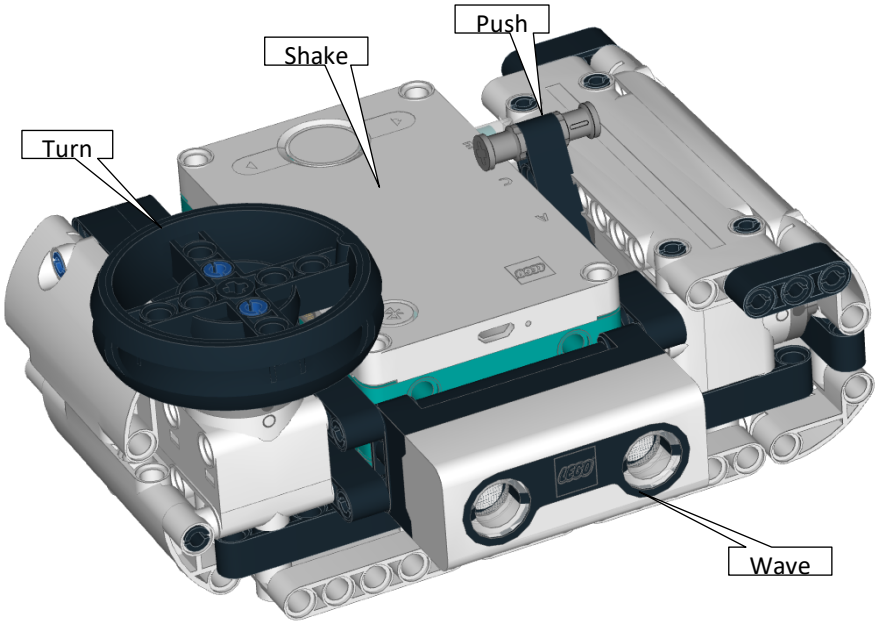

Shake


Turn


Push


Wave

There is only little time to perform the correct move. If the player acts too slow, the game ends with a tone, the central button lights up red. If a wrong move is performed, the button lights up blue. In both cases, the number of correct moves is displayed as the final score.



Change game parameters

While the number of correct moves increases, the time limit is decreased by 10% every 5 moves.

You can change the initial time limit as well as the interval and the rate of time decreases to make the game faster or slower.