Robo Wunderkind Curriculum

Learning is a lifelong and never-ending process for any individual. However, we all know that the most intensive human development occurs during childhood. In our early years, we don't simply learn new skills or gain new knowledge - we discover HOW the world around us works, and how we as human beings work too.

Our ideas, representations, attitudes, understanding, and critical thinking all develop in early childhood. Robo Wunderkind is designed to support the basic needs of children to learn and experience. It helps create a positive attitude to learning and a perception of learning as a fun activity, while simultaneously helping to keep children engaged even if they have a particularly difficult problem to solve. While playing with Robo Wunderkind, children learn by interacting with the real world; they learn by exploring and discovering as individuals.

Robo grows together with your child through complex problem solving, which occurs during the process of coding projects of varying difficulty levels. We will try to make your preparation for lessons with Robo Wunderkind as easy as any game by providing everything you need to start in this curriculum book. Join your students in playing with Robo, and learn and have fun together!

Table of content:

Block 1. Educational concept

Block 2. Robo Wunderkind key info:
- 2.1. Robotics Kit key info
- 2.2. Lessons Key info

Block 3. Tutorials for teachers
- 4.1. Basics and logics of Robo Wunderkind robotics
- 4.2. Basics and logics of Robo Wunderkind coding
- 4.2. Projects 1 - 3

Block 5. Key vocabulary

Block 6. Supporting materials

Educational concept:

Why is Robo Wunderkind a great choice to teach the kids in your class?

Robo Wunderkind MODULES:

- *Fine motor skills;*
- *Spatial thinking - Identifying, analyzing, and understanding the location, scale, patterns, and trends*
- *Modularity - the LEGO™-like concept allows kids build and rebuild the robot any number of times, so there is no fear of failure and no possibility to make things wrong!*

ROBOTICS for kids - future is here!

- *Engineering skills - understanding and using of the principles of engineering to build the robot and to get the particular result;*
- *Understanding of scientific concepts - Even children can make experiments, formulate the theory or invent a new concept!*

CODING - even a child can code!

- *Cognitive skills*
- *Algorithmical, logical, computational thinking*
- *Metacognitive skills such as planning, problem solving and reflexion;*

Now LEARNING is fun and easy:

- *Positive attitude to learning*
- *Intention to self-education;*
- *Rational attitude to the mistakes and critic;*
- *Digital literacy*

Boosts CREATIVITY and ACTIVE playing:

- *Opportunity to create new real and virtual things and to see the real result in a few seconds;*
- *Active physical skill! - the interactions with Robo are active, Children need to move a lot playing with a new friend, so it helps keep them active and healthy.*

Positive SOCIAL EXPERIENCE:

- *Cooperative play;*
- *Opportunity to work in pairs, small groups or even as a whole class-team!*
- *Family activity and new social contacts on the base of common interests;*
- *The possibility to create projects, make the competitions, meetups and many more!*

Robo Wunderkind Robotics Kit key info.

Robo Starter Kit overview

1. What's in the box?

- Main Block
- 5 blocks: Distance Sensor, 2 DC Motors, Servo Motor, Connector Block
- Button
- RGB light
- 7 connectors
- 1 wired connector
- 4 Lego™ adapters
- USB charging cable
- Robo Wunderkind Quick Start Guide
- Stickers

2. What else do you need (device requirements)

- iOS 8 / Android 4.4.2 or newer with Bluetooth 4.0.

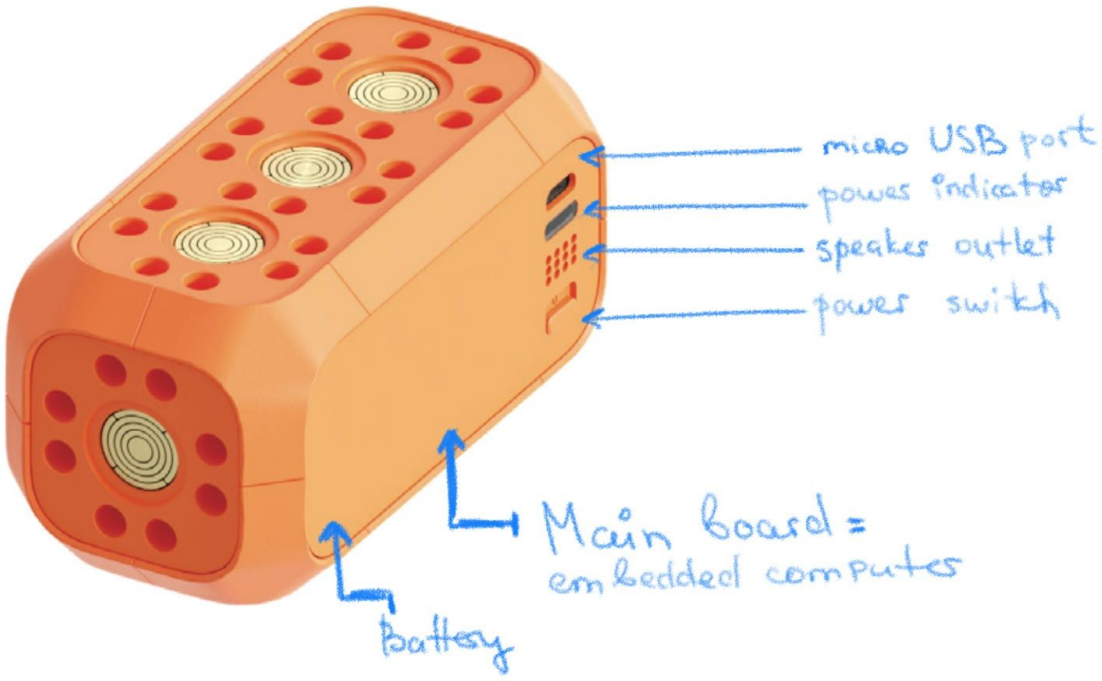
3. Our Apps:

- Robo Code App
- Robo Live App

4. What's next?

- Install the Robo Code on your device
- Turn on Bluetooth on your device
- Open the Robo Code App and follow the special projects / tutorials for teachers to get to know the basic functions of Robo Wunderkind robotics kit

2. Main Block



The Main Block consists of: micro-USB port, power indicator, audio outlet, power switch, main board = embedded computer, battery.

The battery life:

- Lasts for 3 hours total of intensive work
- Better to keep the charging level of the battery at 30% or higher

Switching on:

- Pick up the Main Block, locate the power switch, and turn it on
- The power indicator will light up green; wait a few moments for the start-up sound

Charging:

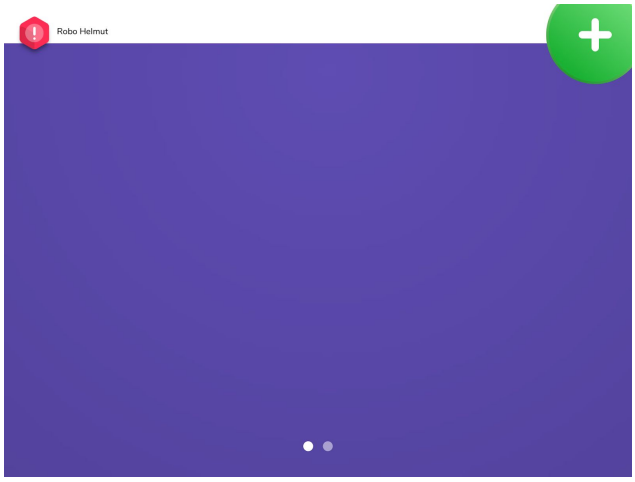
- Plug the micro USB cable to the port at the bottom of the Main Block. The charging sound will be played once you connect the other side of the cable to a power adapter.
- Check the charging status in the App.

3. Firmware update

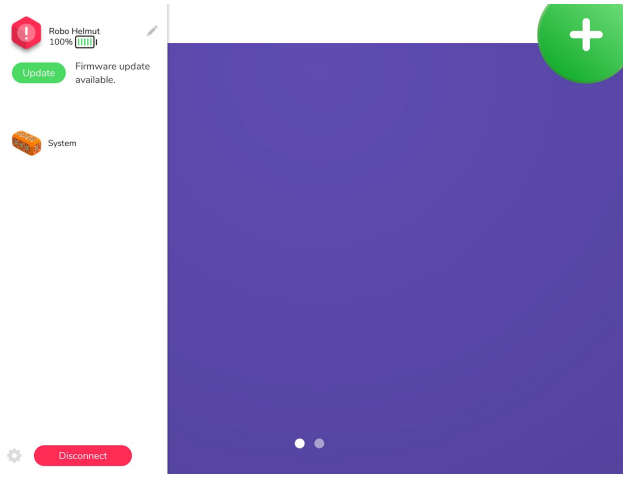
Important:
Make sure you have a stable Wi-Fi connection available. For a safe update process, the main block of your Robo must be plugged in.

Follow the steps:
1. Launch one of the Robo Apps.

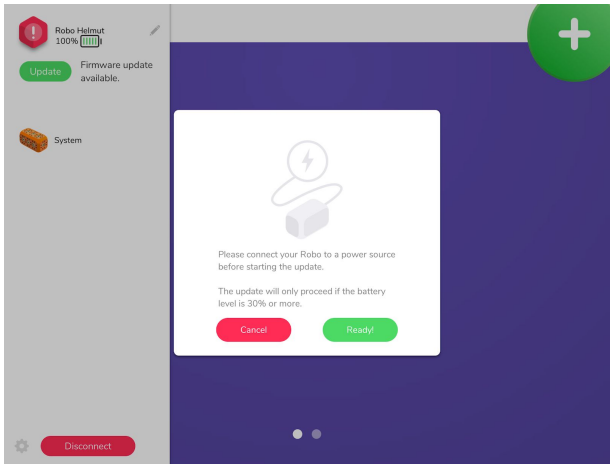
2. Connect to your robot. If there is an update available, you will see the “!” sign on the Robo cube icon in the top left corner.



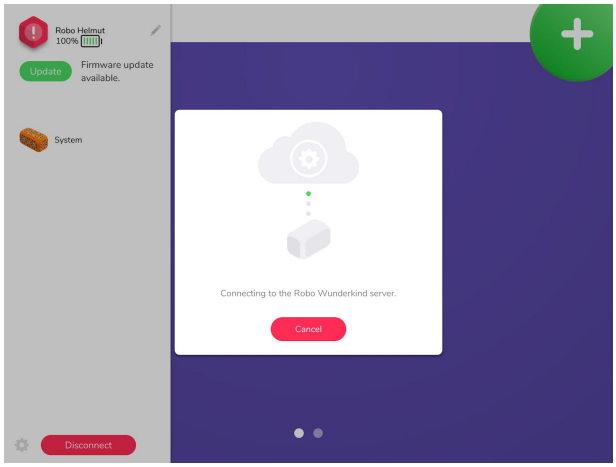
3. Tap on this cube to open the Robot Menu.

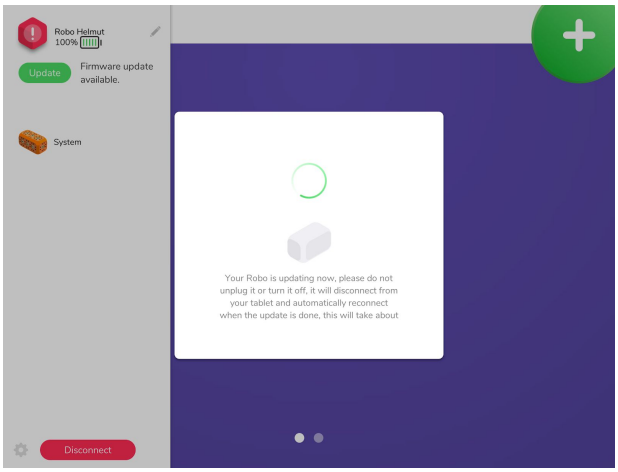


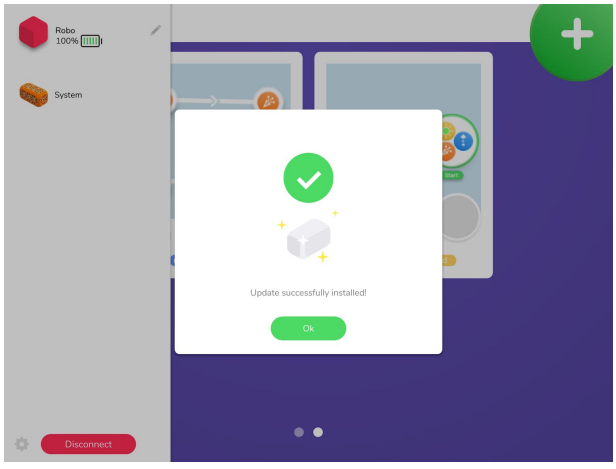
4. Tap on the “Update” button to start the update process.



5. Follow the steps until the update is finished. Your robot will restart and reconnect automatically.







4. Do’s and dont’s

DO:

+ Use the included connectors to attach the modules.
+ Use the included tool to detach connectors.
+ Use Apps to control the modules.
+ Use Lego™ bricks to customize your Robo.

DON’T:

- Use metal or sharp objects to detach connectors.
- allow any contact with water.
- Try to disassemble the modules.
- Throw or drop the modules.
- Rotate motors or servos by hand.

Lessons Key info.

1. Lessons requirements
The App to use: Robo Code App

Subjects: STEM subjects; Robo Wunderkind robotics kit can be used as a supporting tool in other subjects

Students' age: 6-12

Group size: 8-12 students

Recommended time: 60 - 75 mins

Recommended prior knowledge: Students do not need any prior knowledge, as they will learn the basic terminology and the principles of robotics and programming using Robo Wunderkind.

Materials required:
- Robo Wunderkind robotic kit(s)
- Tablet(s)
- Lego™ bricks (optional - to customize the Robo)
- Projector (optional - to show materials for students)

Before each lessons check:
- Robots' battery level
- Tablets' battery level
- The latest firmware and app update

=> Use special **Check List for Lessons** provided in the supporting materials

2. Suggested structure for a lesson with Robo Wunderkind.

To plan:

- Objective
- Learning outcomes
- Focus on the particular module / modules
- Key vocabulary

Structure:

1. **Lead- in:** repetition and activation of the previous knowledge
2. **Storytelling** and story analysis and planning

- 7 - 10 min

3. **Learning by doing:**

1st project / set of challenges:

a. Unboxing
b. Building
c. Programming: different tasks solving
d. Programming: final Code = problem / task is solved
e. Reflexion of the 1st project

- 15 - 20 min

2nd project set of challenges: modification for older students (optional)

a. Problem situation setting: storytelling; story analysis and planning
b. Re-building the Robo
c. Programming: different tasks solving
d. Programming: final Code = problem / task is solved
e. Reflexion of 2d project

- *optional time*

4. **Own project** on the basis of the learnt skills:
    a. Task(s) setting
    b. Building and coding (independently / in pairs / in small groups)
    c. Presentation of the project and feedback

- 15 - 20 min

5. **Final reflection & feedback**

- 5-7 min

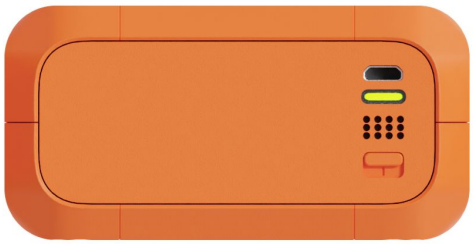
The total time of each lesson: 40 - 60 min

3. Rename your Robo:

Attention! Each Main Block has a name, which is set by default as “Robo”. To connect the Main Block to the Robo Code or Live App, you need to know the name of the particular Main Block. To avoid confusion, please rename all Main Blocks so that they each have a different name. We advise using numbers / names and special stickers for Main Blocks to put the numbers / names on them. You will find the stickers for naming the Main Blocks in Supporting Materials.

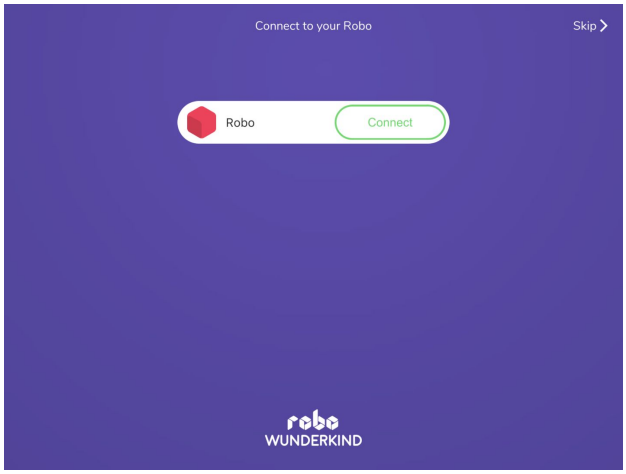
How to rename:

1. Switch on the Main Block

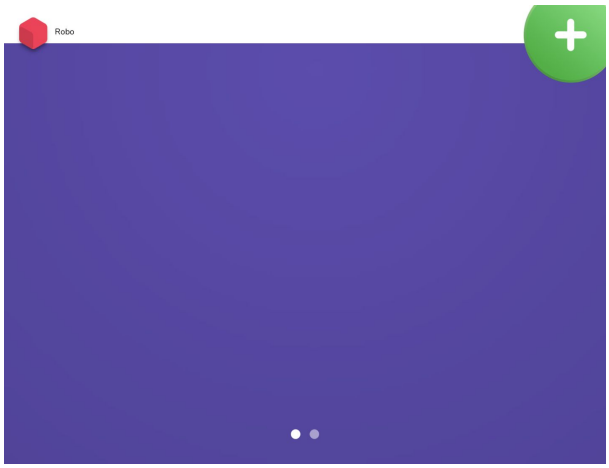


2. Open the Robo Code App

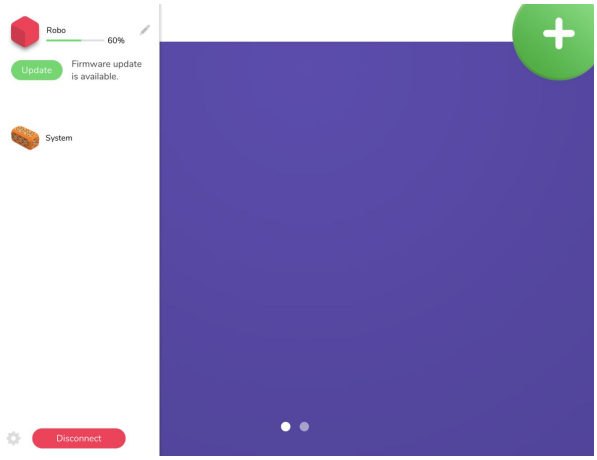
3. Connect the Main Block to the App



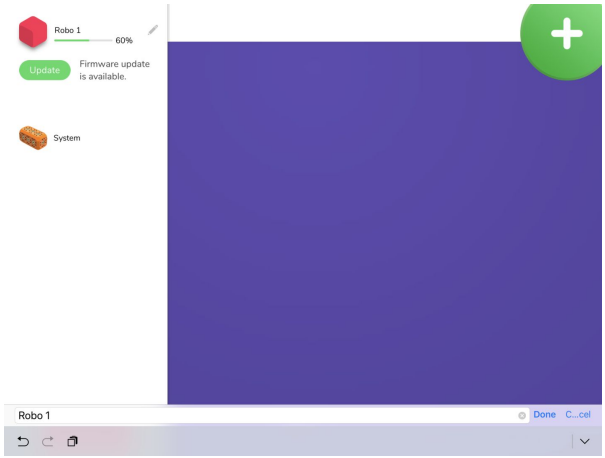
4. In the left corner of the Project Screen you will see the red cube - Robo Menu with the name “Robo” on it



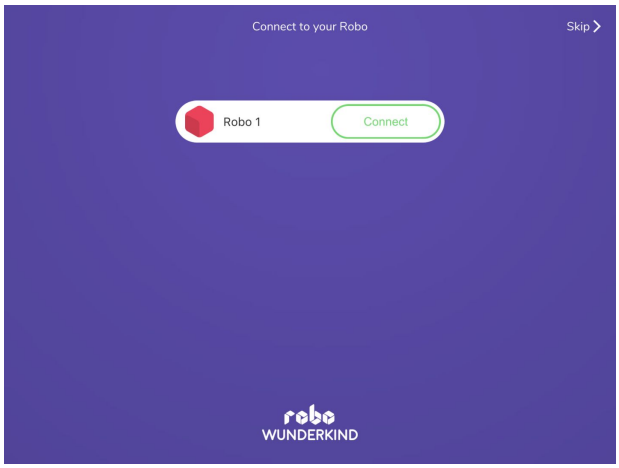
5. Click this button and you will see the information about the Main Block: its name, battery level, and connected modules



6. Click the pencil icon and **rename the Main Block**



7. Restart the Robo Code App to see the changes
8. Don’t forget to put the same name on the Main Block using the sticker.



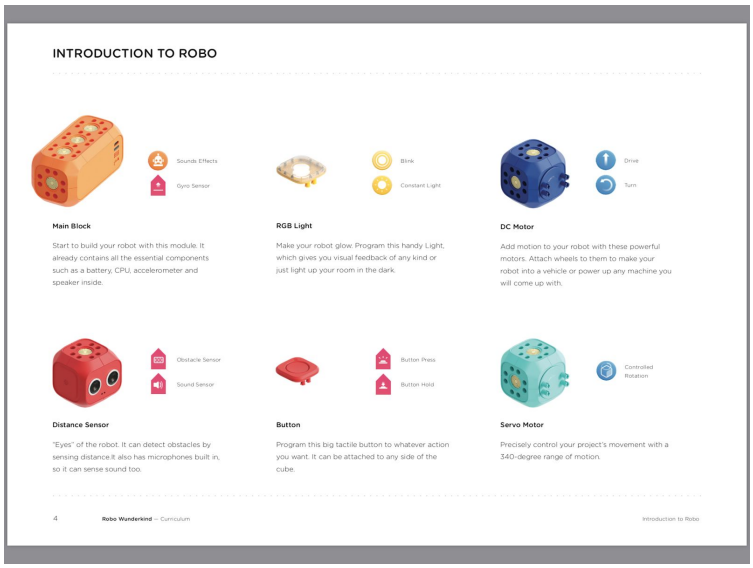
Tutorials for teachers.

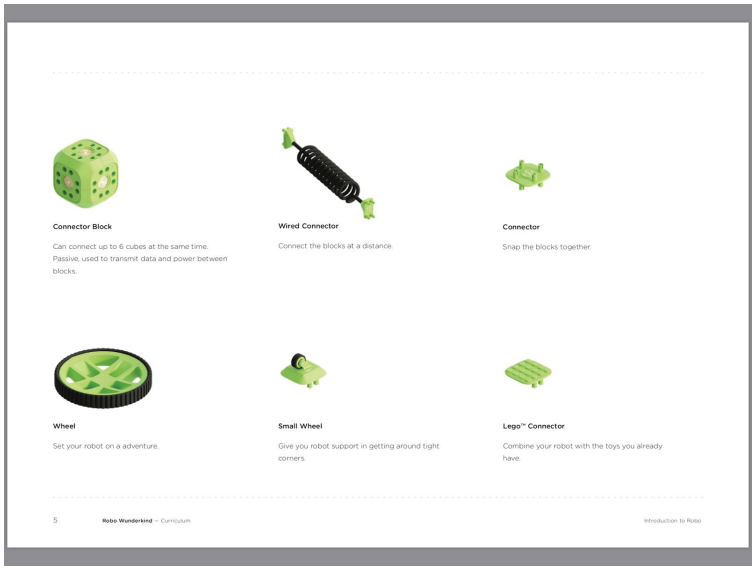
Let's imagine ourselves being kids and play, discovering the main functions of Robo Wunderkind robotics kit. Build the robot and programm different codes! Go through our tutorials for teachers step by step exploring the Robo. Be a kid and have fun!

Here you will find the basics of Robo Wunderkind **robotics** and **coding** and **3 very fast projects**. Take only 30 minutes and become a master of Robo Wunderkind!

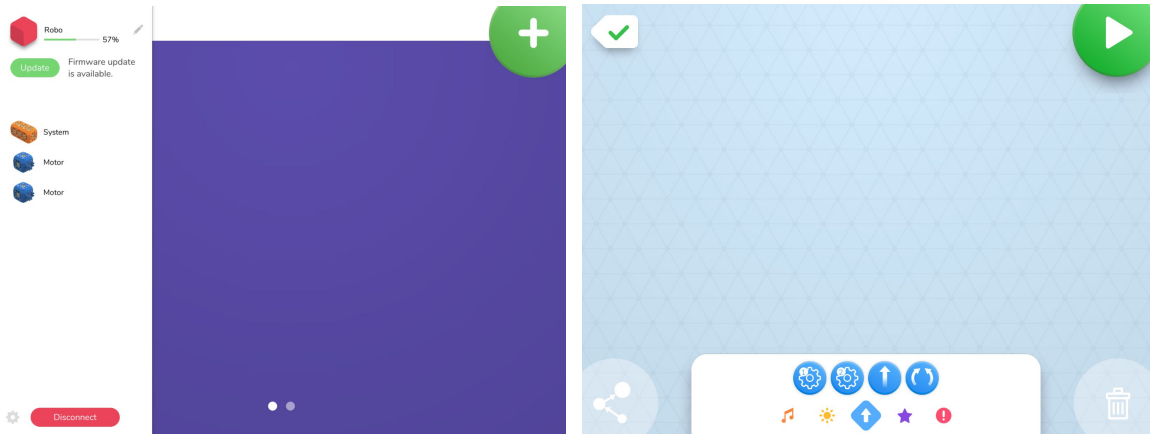
1. Basics and logics of Robo Wunderkind robotics

1.1. Robo's Modules have **different colors**, each expresses a block's functions and is related to the colors of actions in the Robo Code App. It makes coding more intuitive for students.



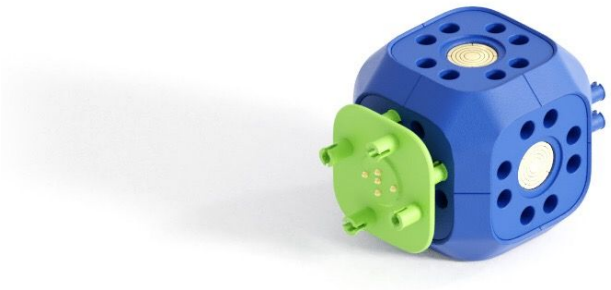


1.2. To **program** a certain action, you need to be sure that the particular module for this action is assembled to your Robo's Main Block.



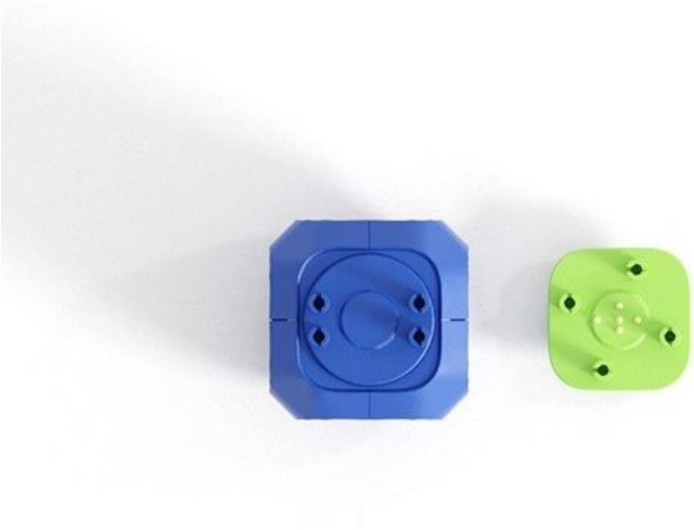
1.3. To make **modules work**, you need the electricity from the Main Block, which passes through small **Pogo pins on connectors**. There are some aspects to be aware of:

- The connectors need to be assembled properly, with the sides parallel to the modules' sides



- Because of the tight connection, in order to disconnect the modules you need to use the **disconnecting tool**

- Some of the modules such as two DC Motors and Servo Motor **do not have Pogo pins.** One side of these modules consists a rotating part, that is made out of plastic, so the electricity doesn’t flow through it.



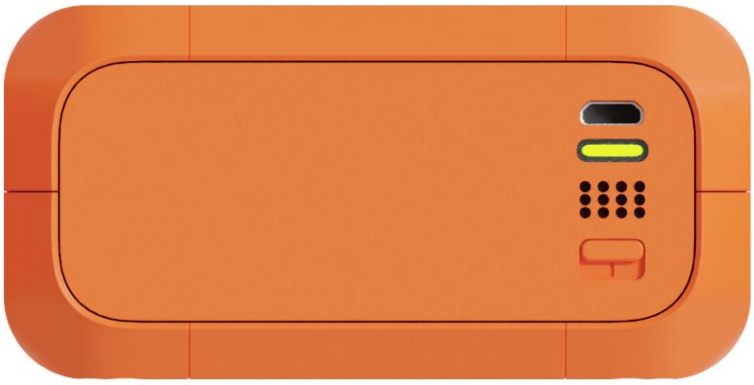
- The module that is attached to the moving part of Servo Motor **doesn't provide electricity**, therefore, it **will not communicate with the Main Block**.

- **Wired connector** helps to resolve this problem. You can connect the module you need to work with the Main Block directly by Wired Connection.

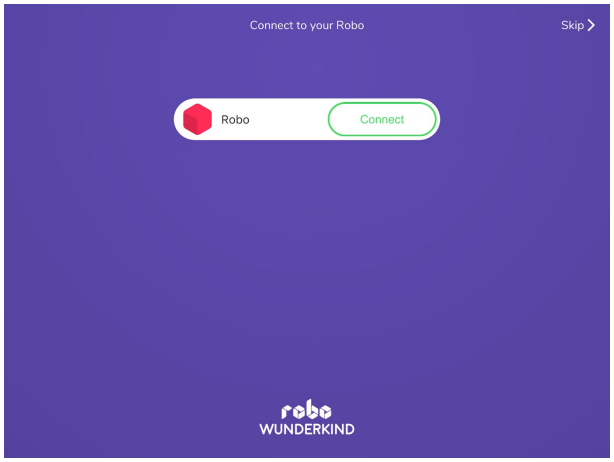
2. Basics and logics of Robo Wunderkind coding

How to connect to the App:

- **Switch on** the Main Block

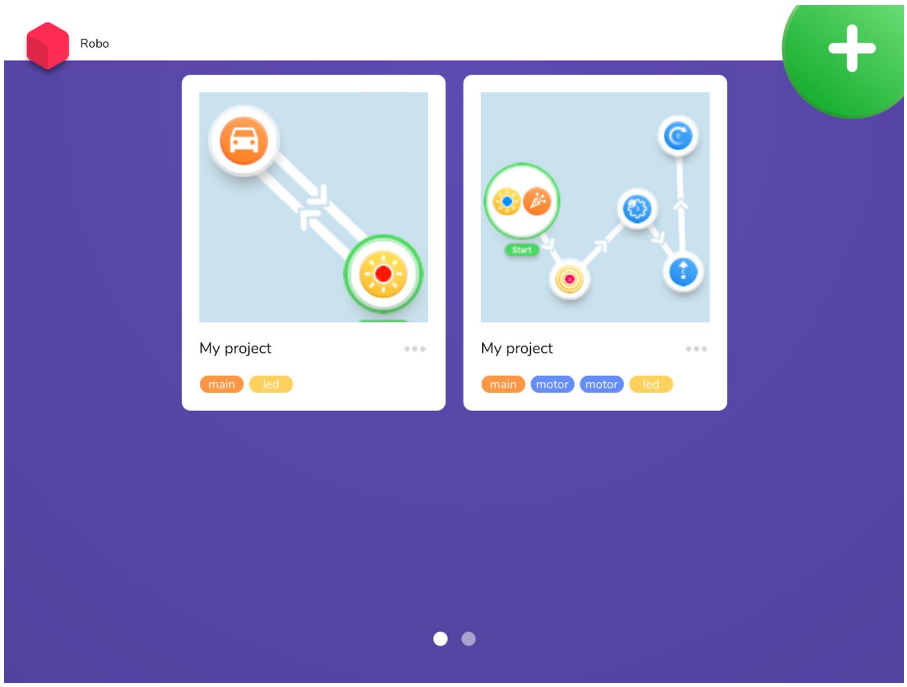


- Take the tablet and check if the **Bluetooth is on**
- Open the **Robo Code App** and connect your Main Block to the tablet

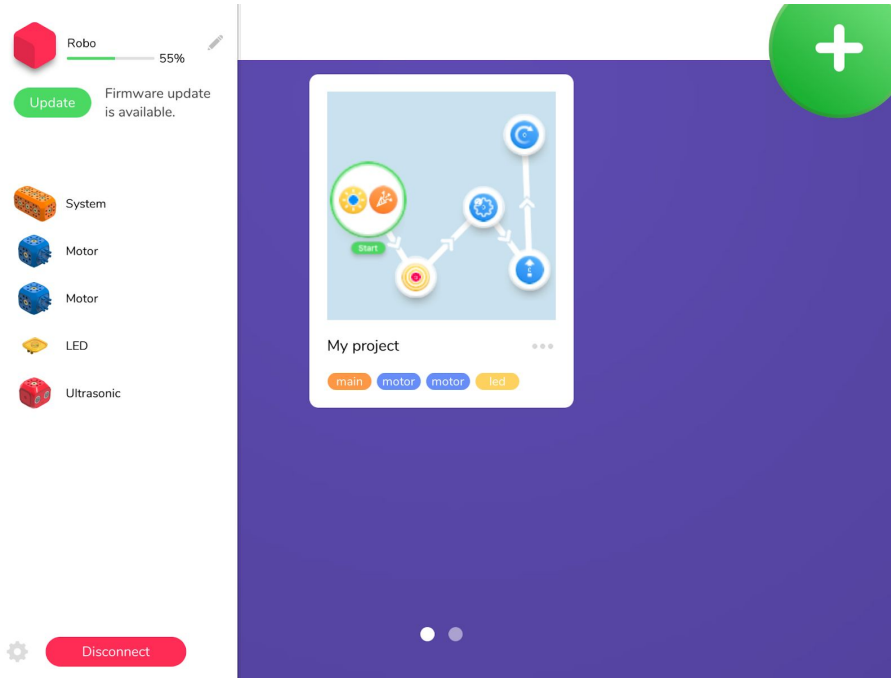


The Robo Code App: What's on the screen?

Projects Screen - here you can see the projects you create



Robot Menu in the left corner of the screen - here you will find the information about your Robo: its name, battery level, firmware update and the module connected



Start Button: click the green plus button to start a new project.

Programing Screen:

Actions Dock: here you will find the actions to program sounds, lights, movements, specials, and conditions.

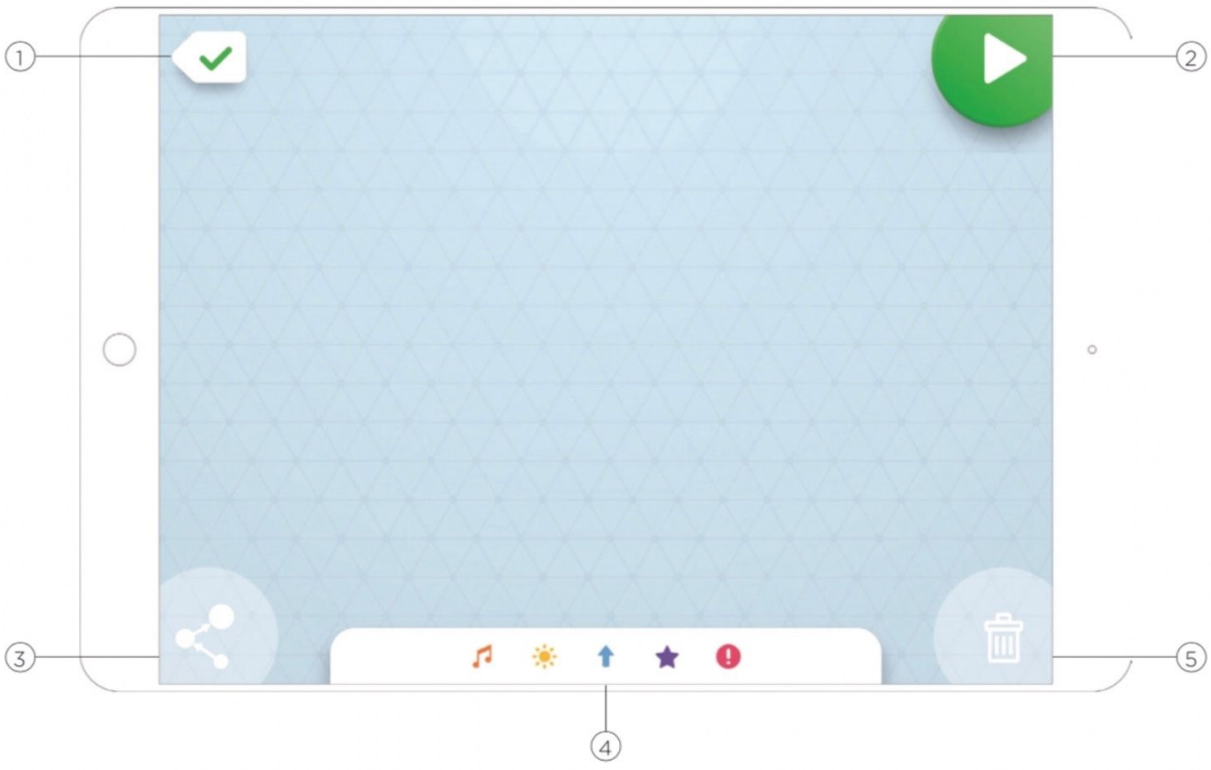
Connection button: use it to connect the actions into the code; be aware of direction of connection - transition between the actions.

Trash bin: use it to delete the actions or connections.

Play button: try your program out!

Stop Button: when you run your code the Play Button changes to the Stop Button; you can stop your Robo each time you want. **Attention**: don't forget to stop the program from running before you make the changes in the code!

Save & Exit button: click it to return to the Project Screen; don't worry, your project will be saved automatically



=> Now you know the logic of Robo Wunderkind robotics and the interface of Robo Code App. You are ready to program!

Project 1.
There are 3 steps in the very first project:

Step 1. Focus: Main Block <> Program Sounds

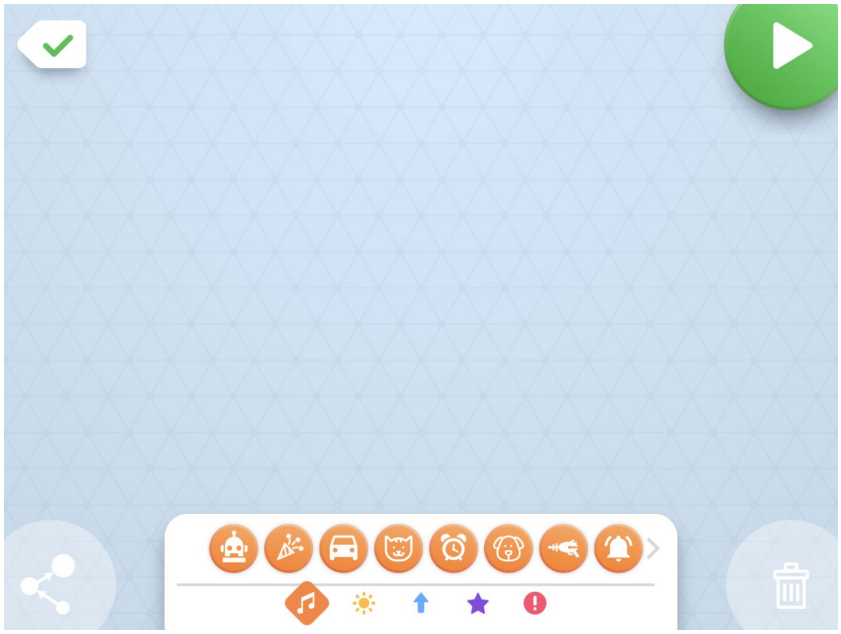
For the very first step you need only the **Main Block**. It contains not only a battery and CPU, but also has the speaker and accelerometer inside. Using just the Main Block, you can already programme the **Sounds**.

The **Sounds** are **orange,** as is the **Main Block,** and are located in the first action in the Action Dock. Click the orange note-icon to see all sounds available. Let's try to program the first code using only the Main Block.

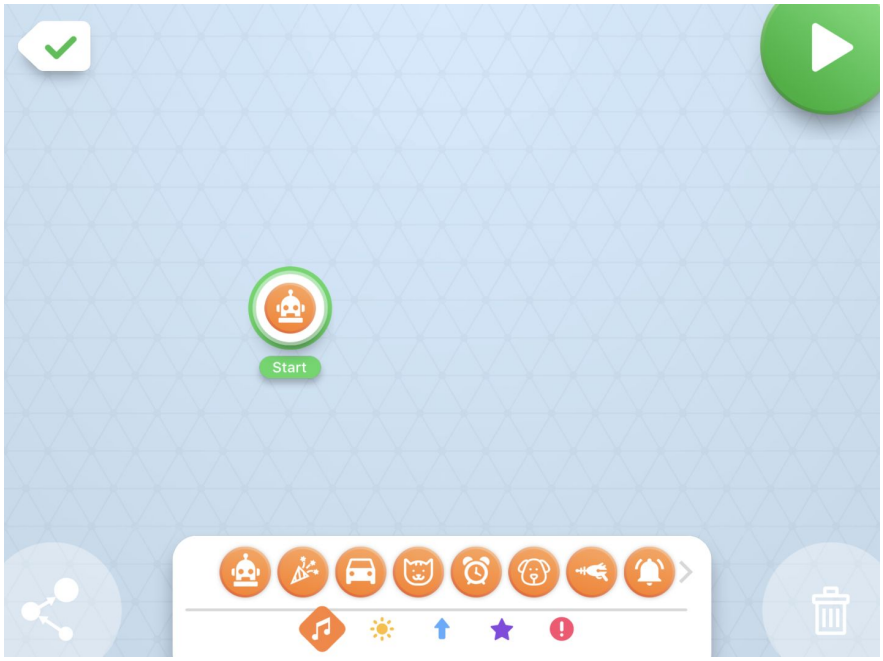
Build:

Program:

- Make a new project by pressing Plus Button.

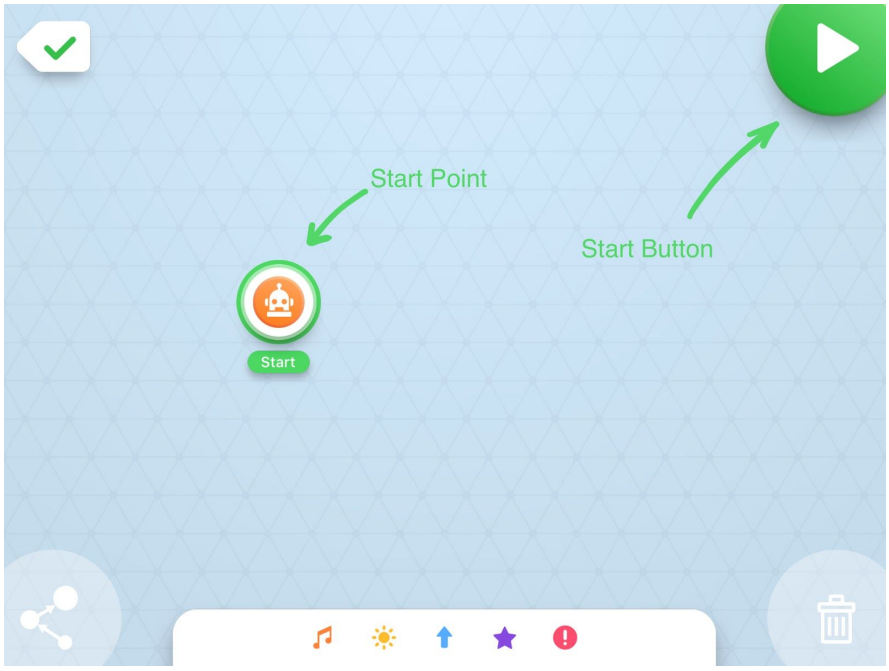


1. Drag the first **Sound** onto the **Programming Screen**; you can place it anywhere you like.

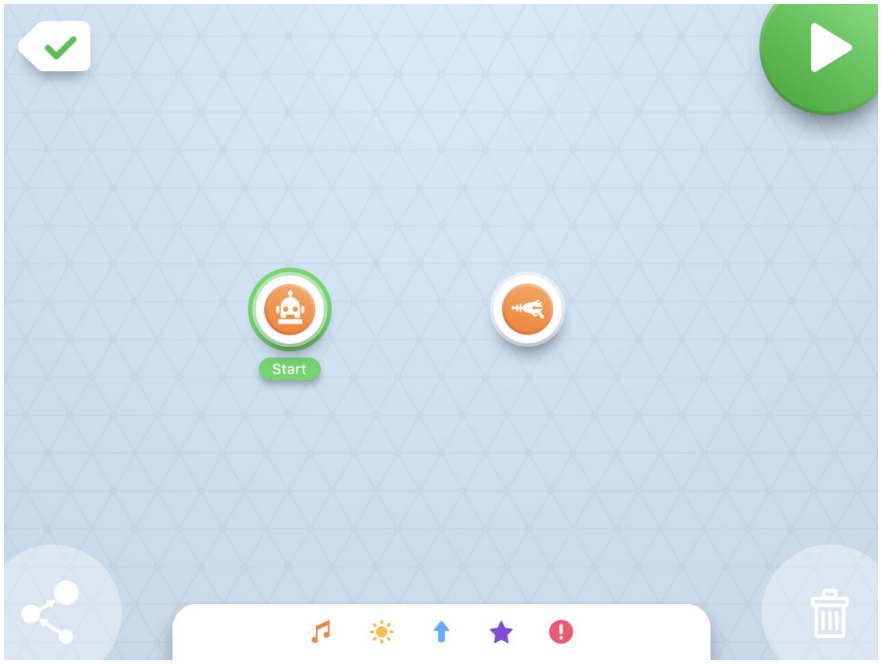


2. The green circle marked with the **Start Point** around the sound means that the code starts here.

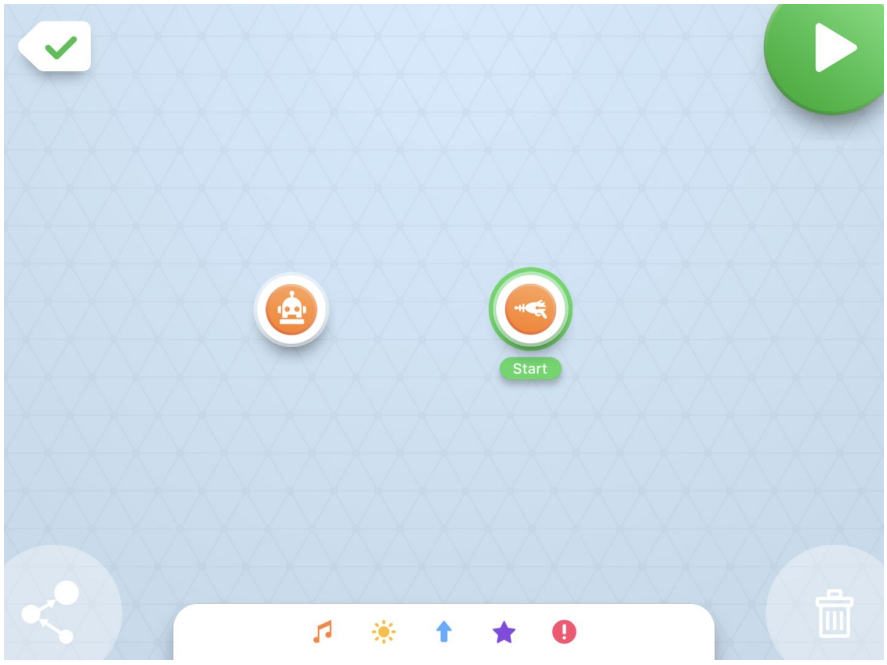
3. Click the **Start Button** to try out your first sound.



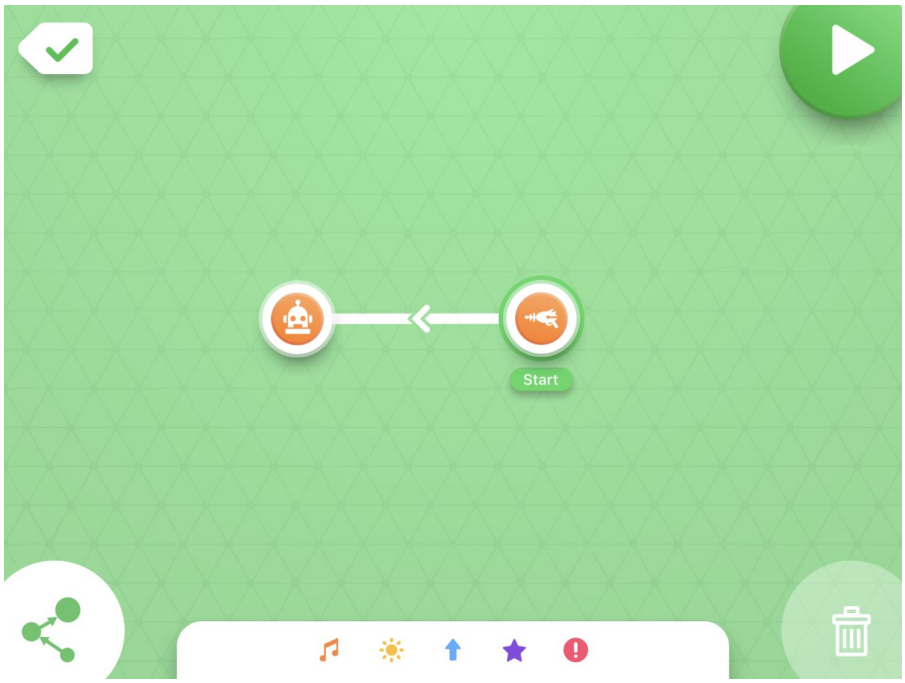
4. Drag the second **Sound** onto the **Programming Screen**.



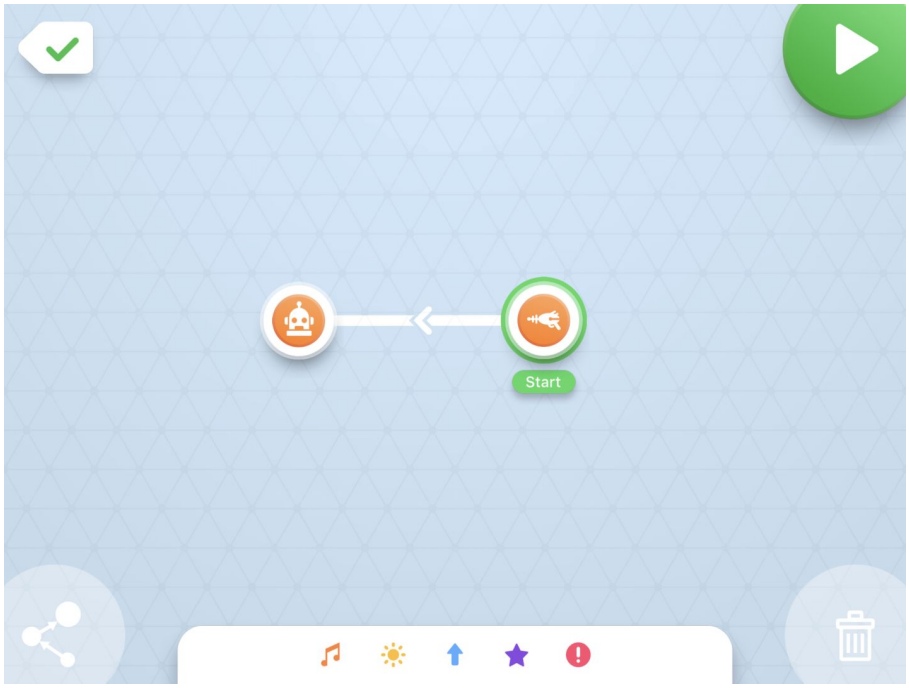
5. To try it out independently, you need to **move the Start Point** from the first Sound to the second Sound; drag the small Start Point and drop it in the middle of the second Sound; try it out!



6. Now let's connect two sounds! Click on the **Connection button. When** the Programming Screen turns green, it means you are in the Connection mode. Draw the arrow / transition, paying attention to the direction of arrow / transition you draw.



7. Click on the Connection Button or on the Programming Screen to turn the Connection Mode off. The Programming Screen will turn blue again.

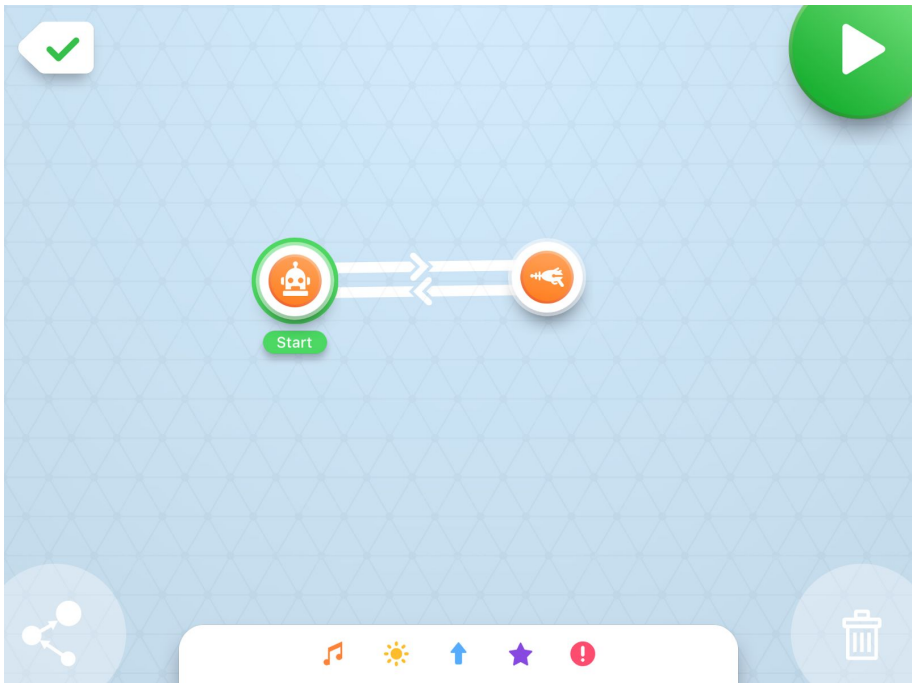


8. There is your first **Code** - two actions run one after another. Try it!

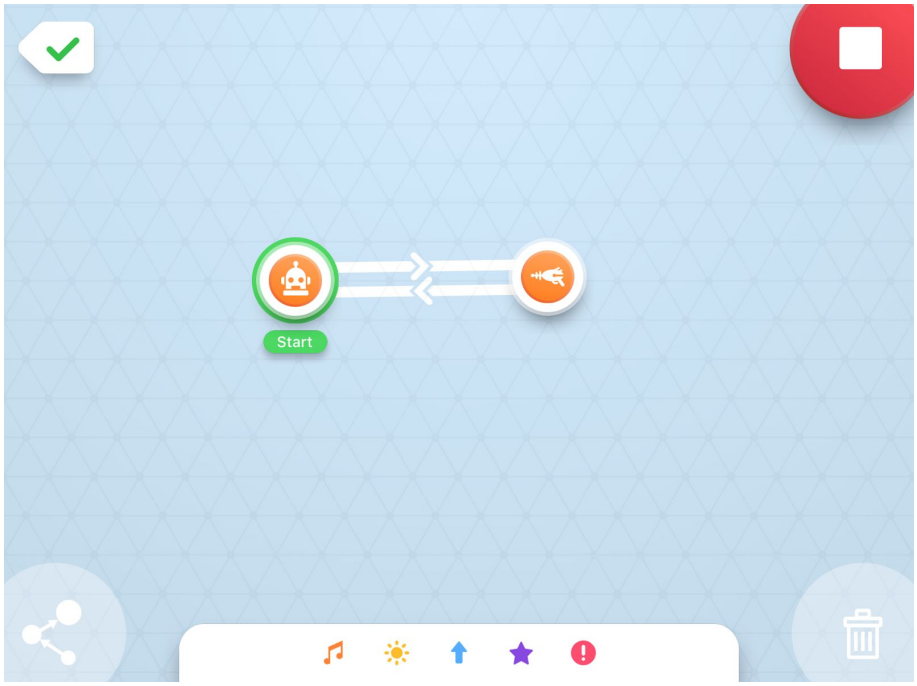
9. **Play around**: drag, connect, and try more Sounds!

Make a Loop: It's time to make a code that runs forever!

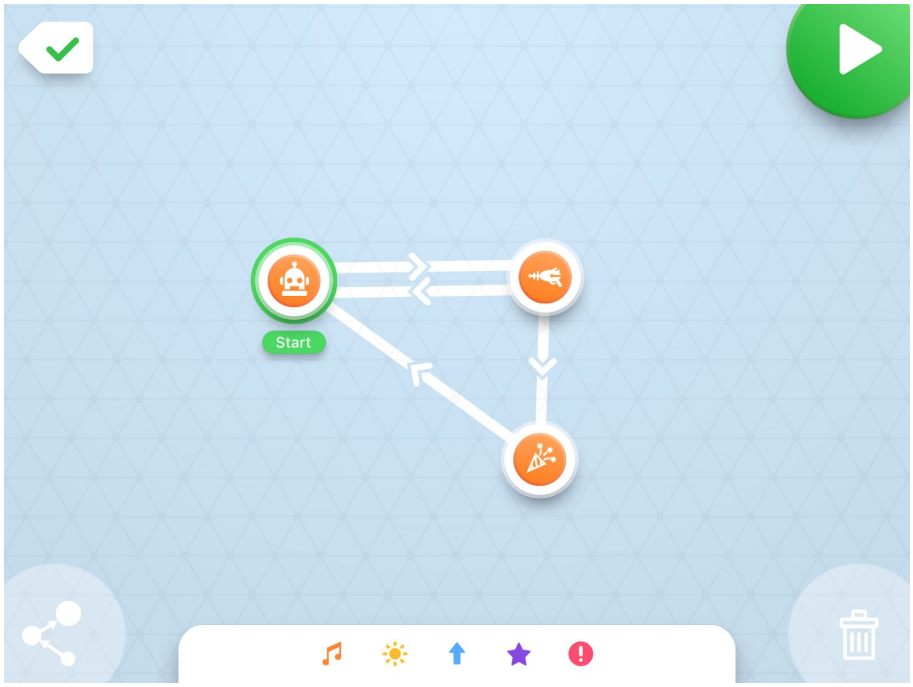
1. Connect two Actions in **both directions.**



2. Run the Code. It will continue playing until you stop it by clicking the **Stop** button.

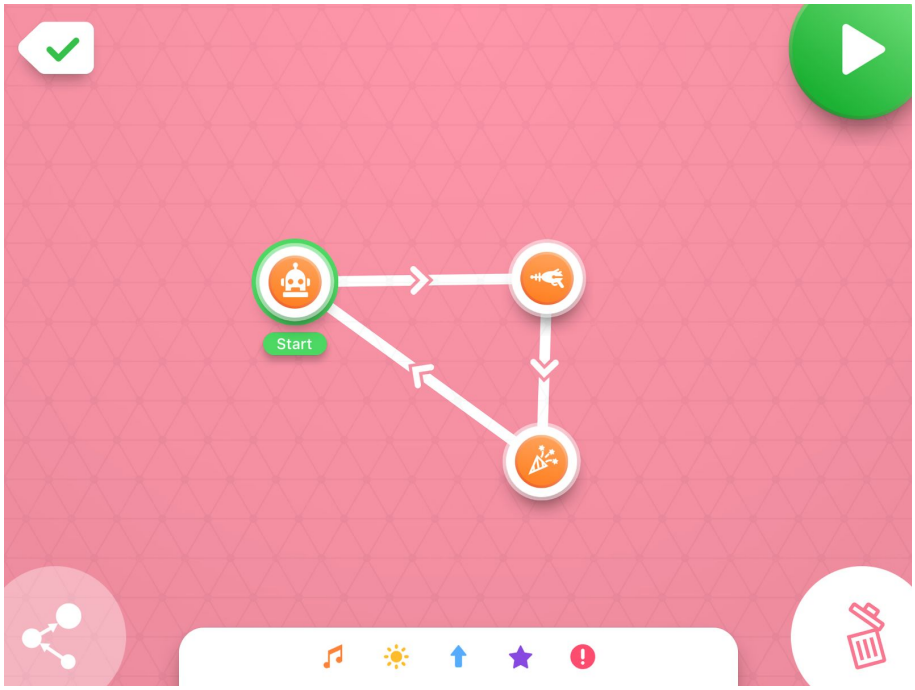


3. Program **Loop** for 3 Sounds

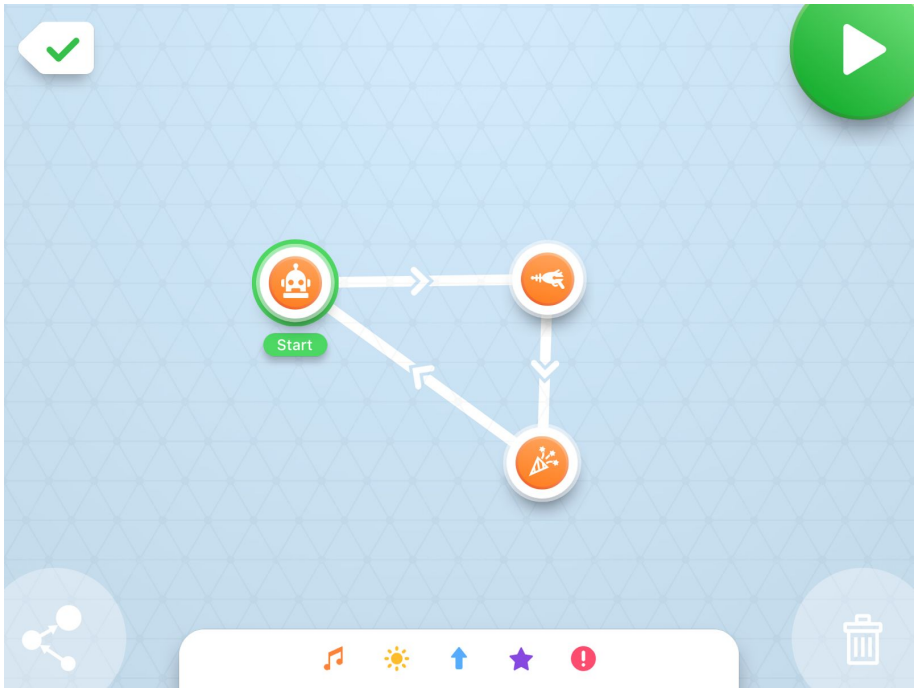


Use **the Trash Bin** if you need to delete Transition:

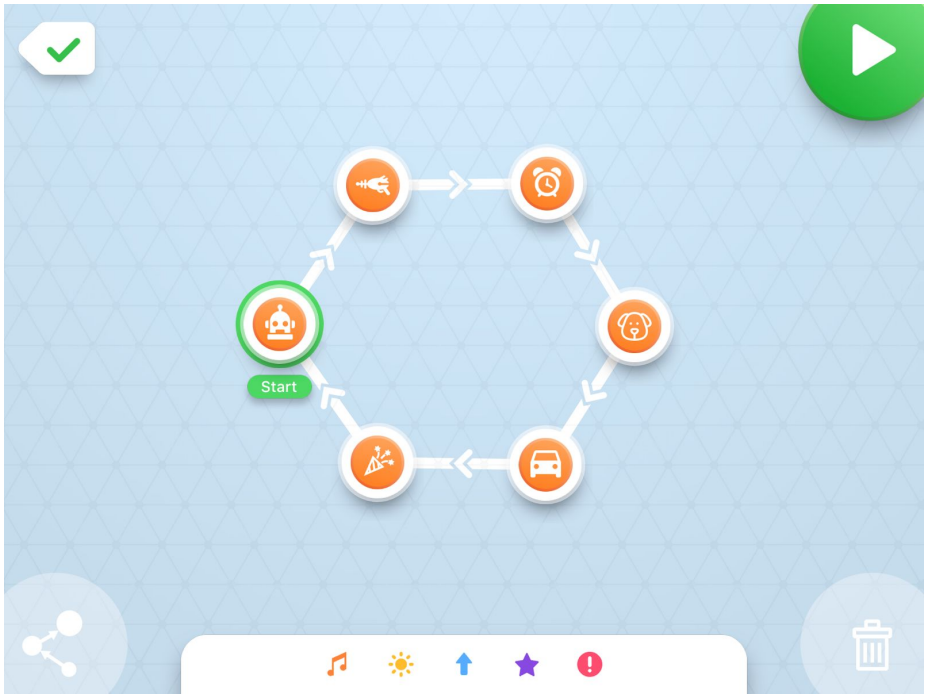
1. Click on the Trash Bin once and the Programming Screen will turn red, which means you are now in delete mode. Click on the Transition you want to delete.



2. Click on Trash Bin or on the Programming Screen once more to switch delete mode off. The Programming Screen will turn blue again.



Play around: program Loop for 4 or more Sounds!



STEP 2. Focus: RGB Light <> Constant Light and Blink

Connect the **RGB Led** to the Main Block. Now you can program two new actions: **Constant Light** and **Blink**. They are yellow, as is the RGB Led module, and are located in the dock as the sun icon.

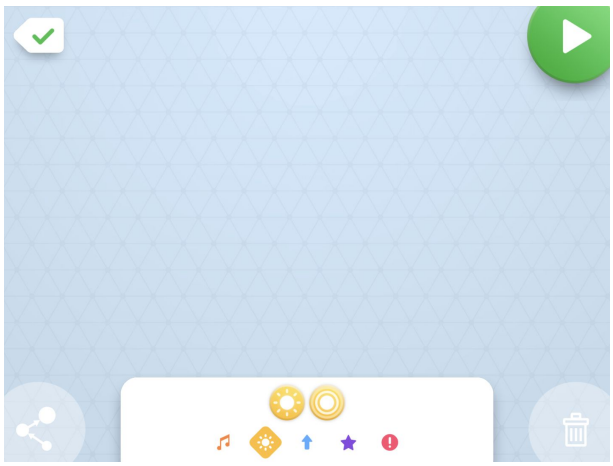
Build:

Program:

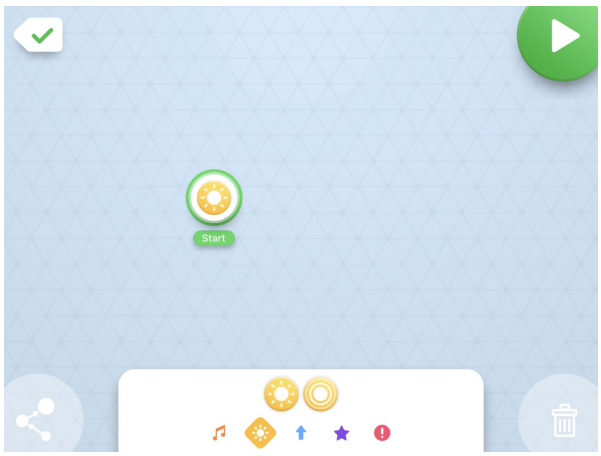
- Make a new project by going back to the Project Screen and pressing the plus button again;

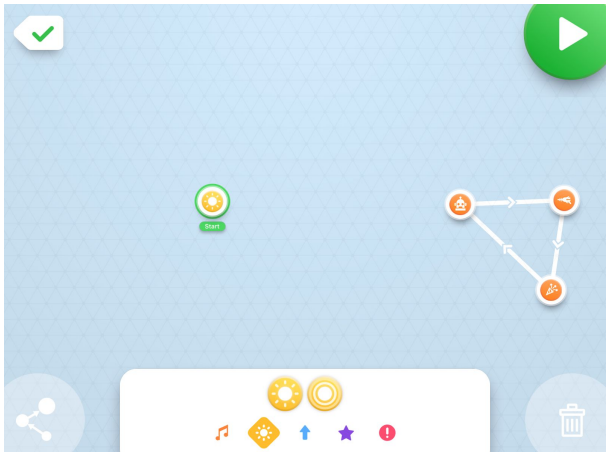
or

- Continue programming in the same project. You can simply put the precious code on the side of the screen, zoom out if you need more space or even delete all Action you don't need.

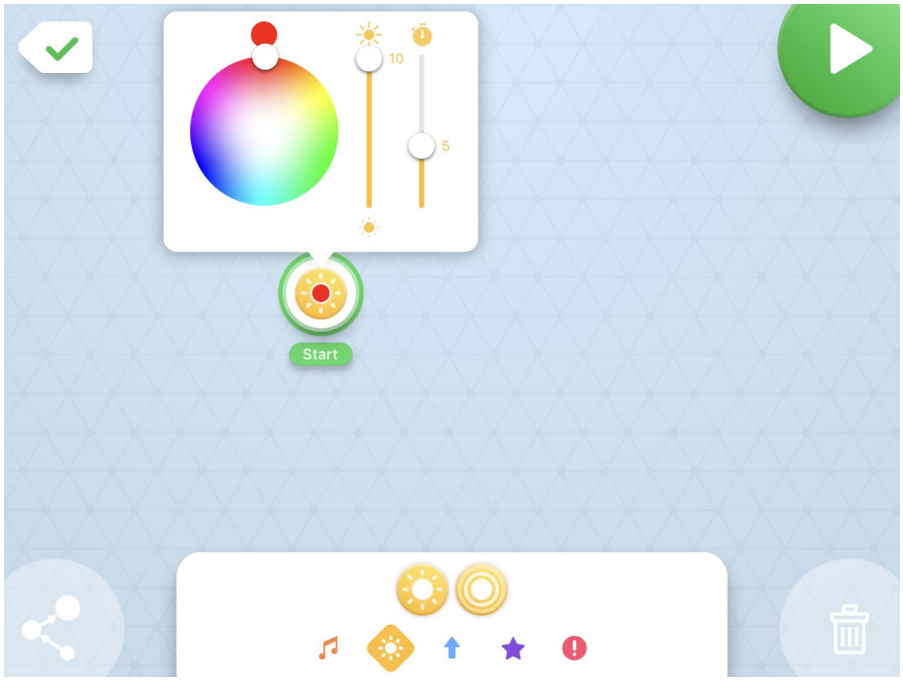


1. Drag the first **Constant Light** on the Programming Screen.

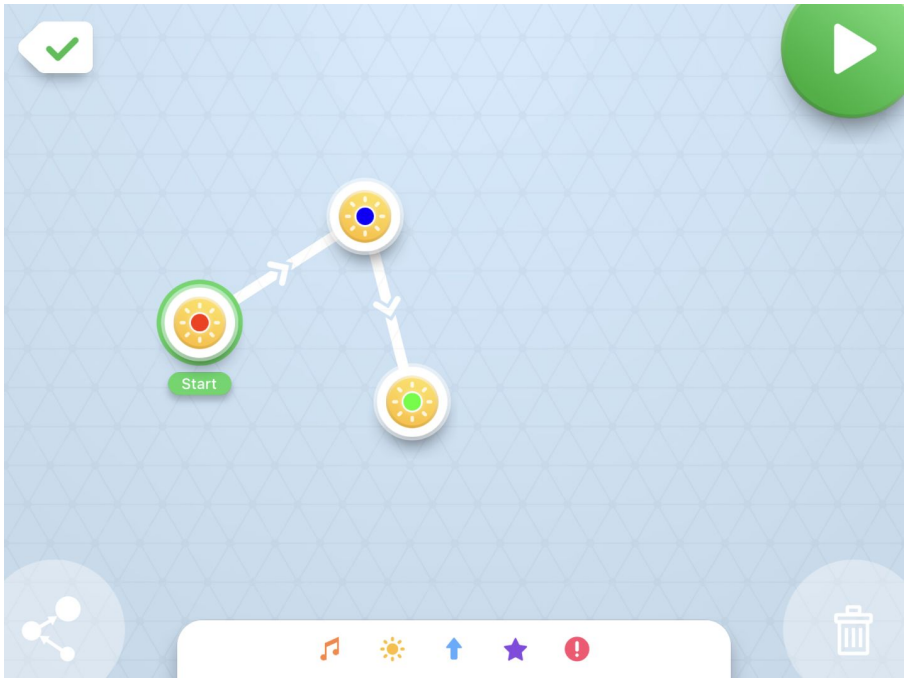




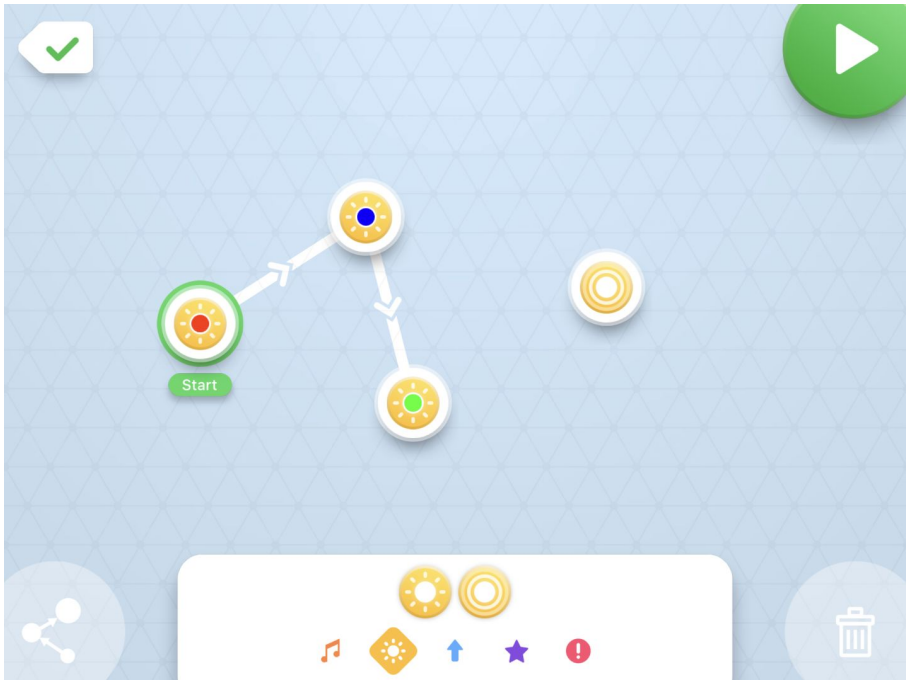
2. Set the **color** and the **time** for the first Constant Light; click on it to open the menu, and click on it a second time to close the settings. Try it out!



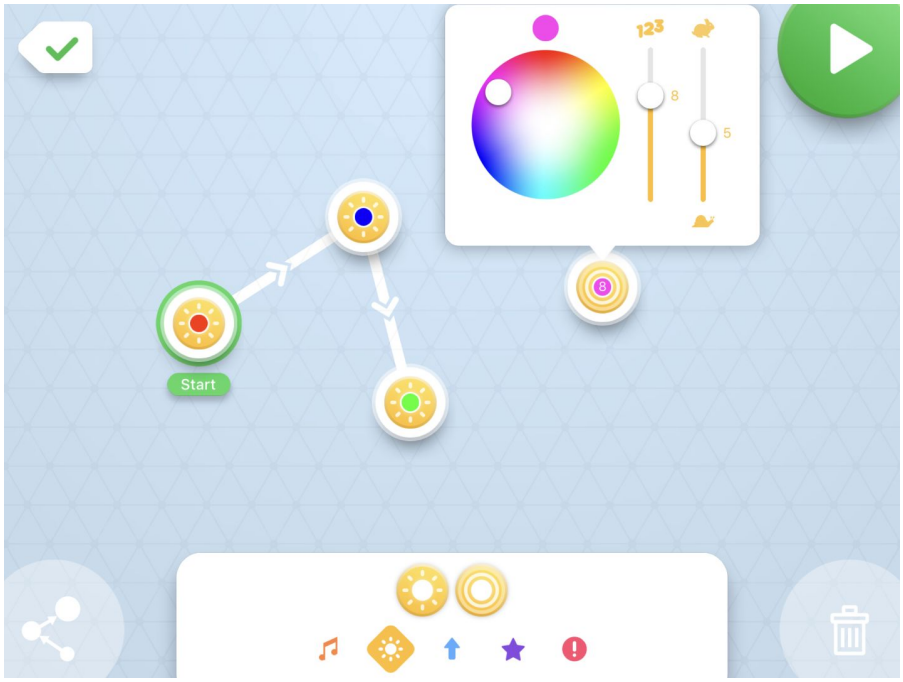
3. **Play around:** drag, set and connect more light actions.



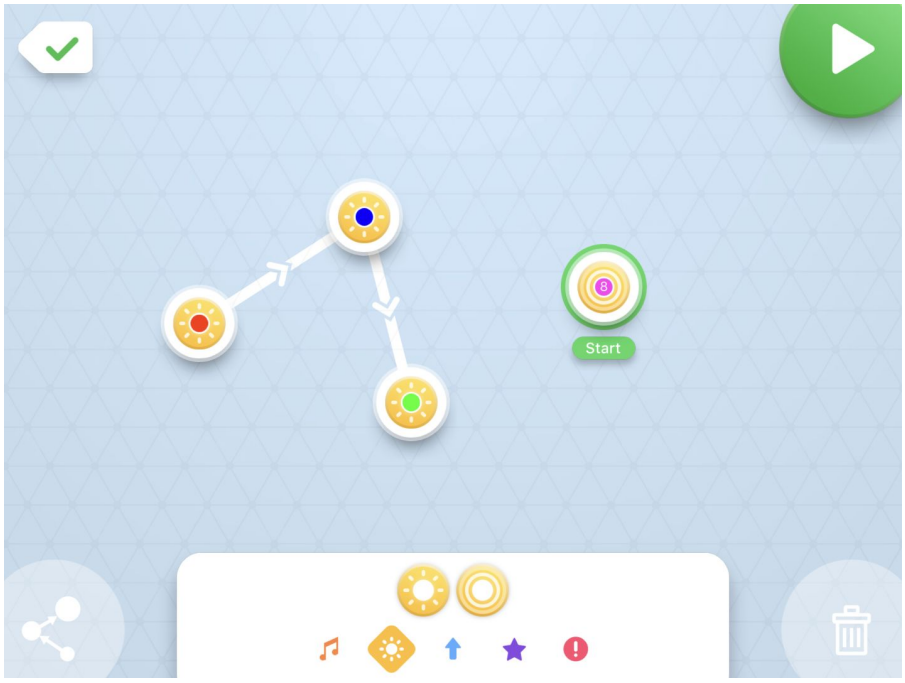
4. Let's try Blinks! Drag the **Blink** on the Programming Screen.



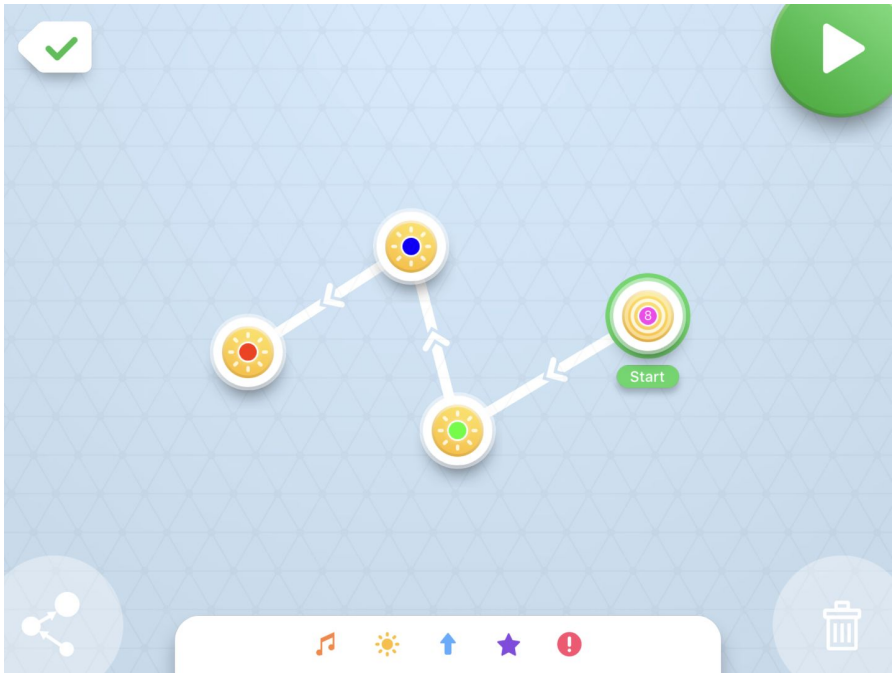
5. Set the **color, the number of times, and the frequency.**

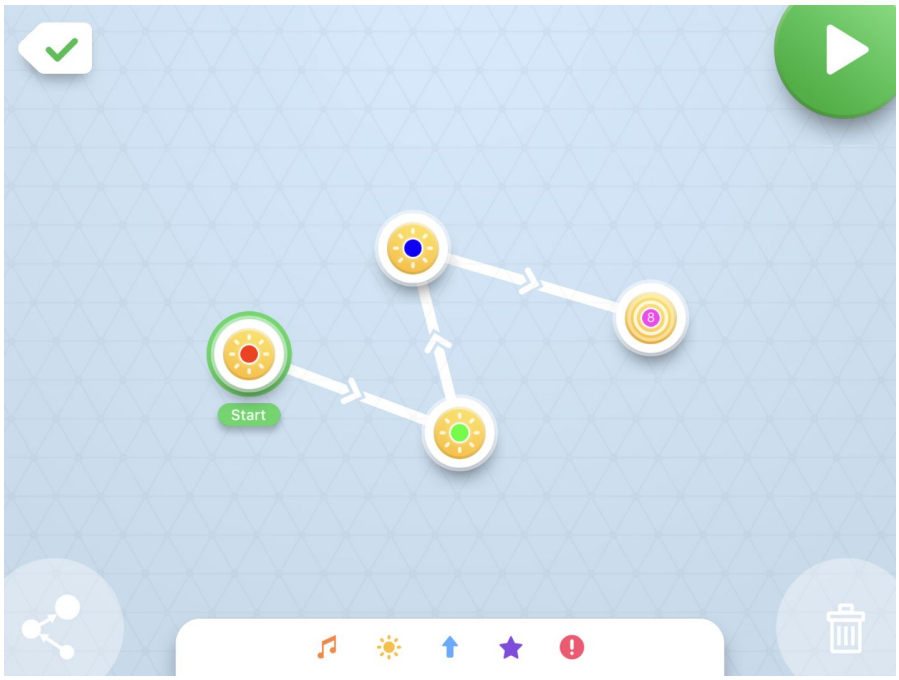


6. To try it out independently, you need to move the Start Point to it.

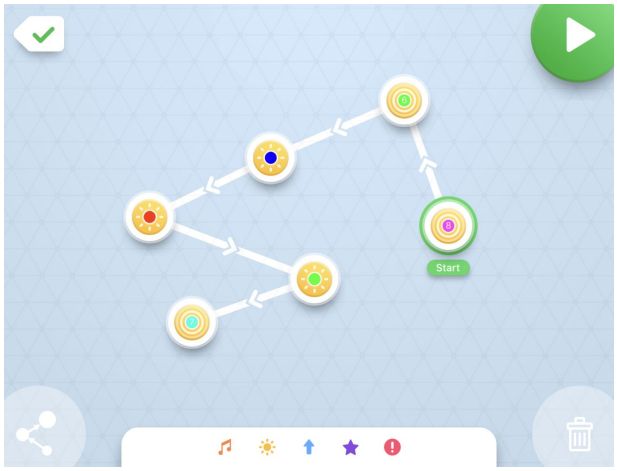


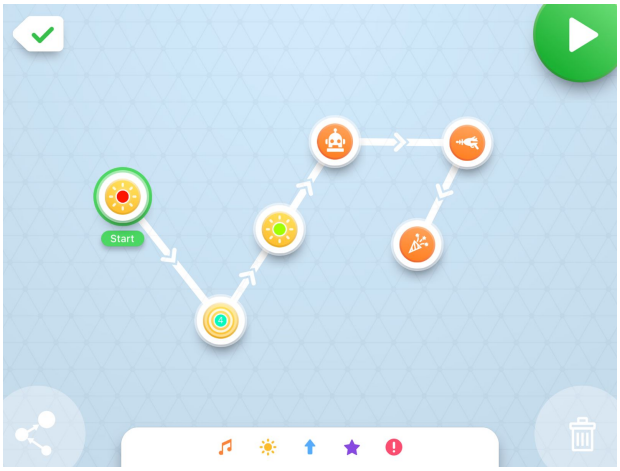
7. Add your **Blink** to the previous code. Change the **Start Point** or **Trash Bin** if you need. Pay **attention** to the directions of arrows / transitions. You can try to do it in two different ways and see the difference!



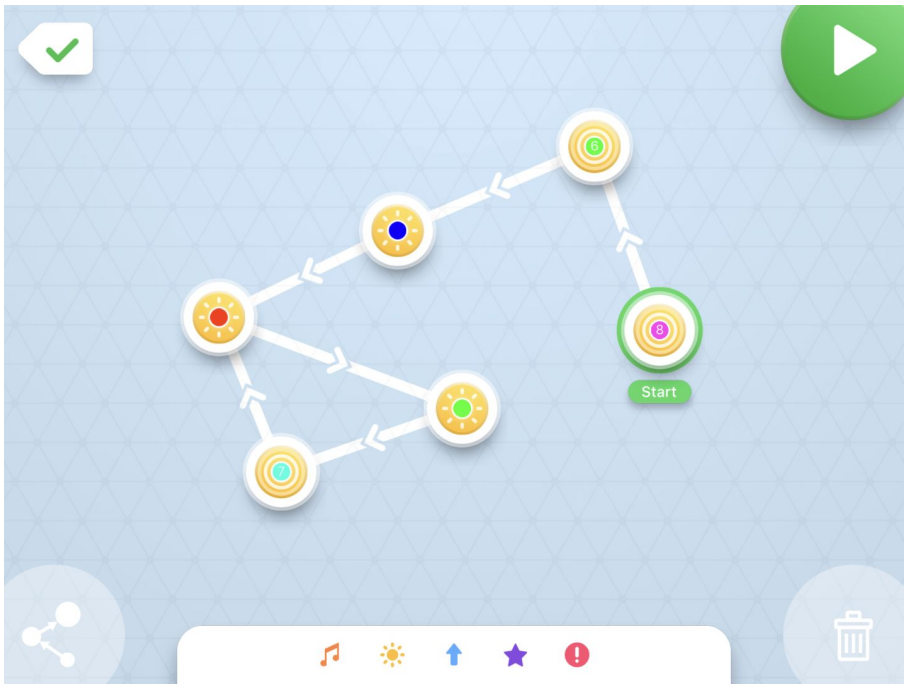


8. **Play around**; drag, set, and connect more **Constant Light** and **Blink** actions or add some Sounds to them.





9. Make the **Loop**! Will the whole code repeating in our example? Which part of the code will be repeating?

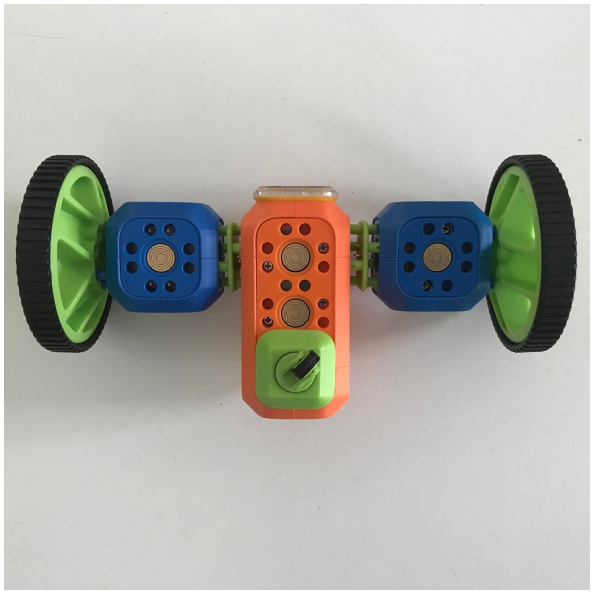
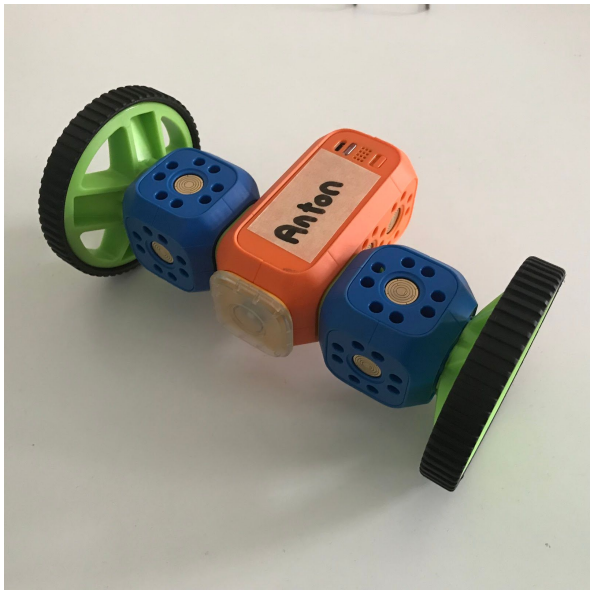


Step 3.
Focus: DC Motors & Wheels <> Motor 1 and Motor 2, Drive, Turn

Let's program Robo to move around! Connect 2 DC Motors to the Main Block and attach Wheels to them. Attach the **small wheel** to build a Robo that can drive.
In this variation with attached two DC Motors you can program movements such as **Drive** and **Turn**, and also program **Motor 1** and **Motor 2** independently. **Drive**, **Turn, Motor 1**, **and Motor 2** actions are colored **blue** and located in the dock as the arrow icon.

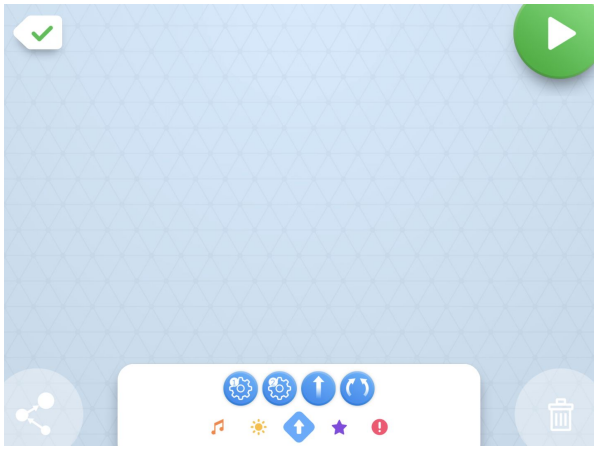
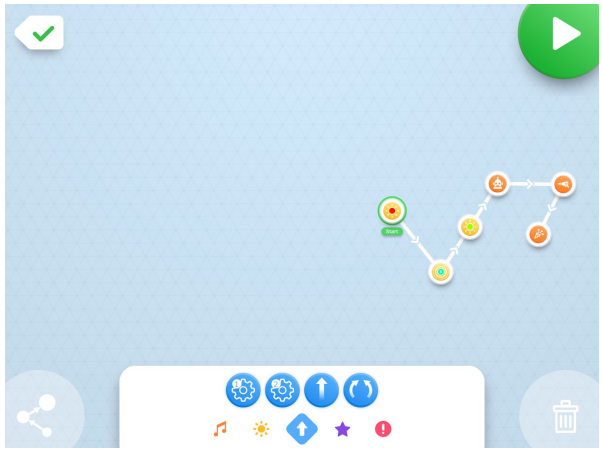
Attention: later you can build with your students any variation of Robo you like, but be aware of the fact that the way in which the DC Motors are connected to the Main Block influences the way Robo will move and turn.

Build:

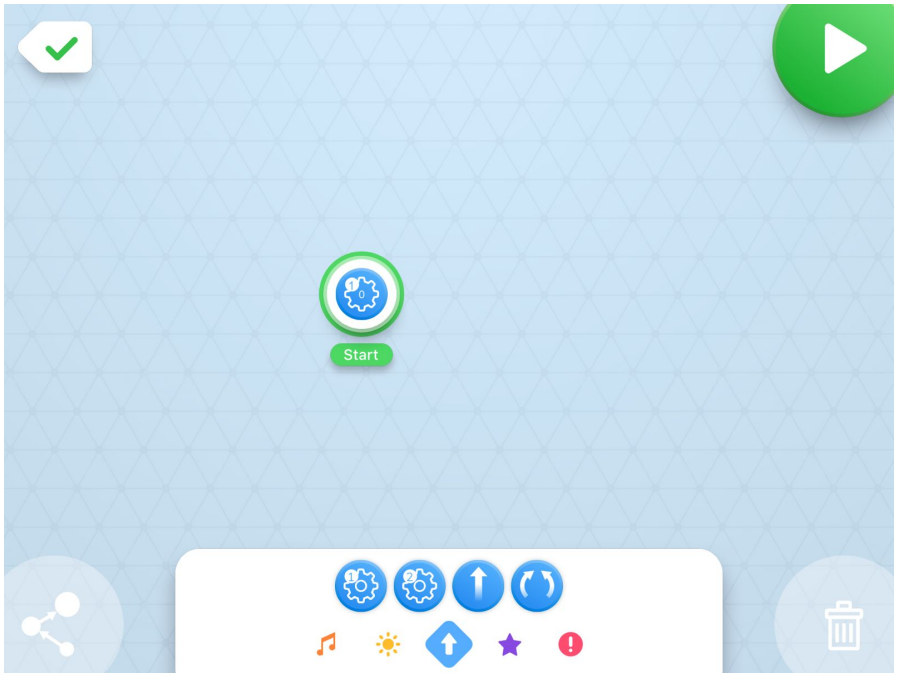



Program:

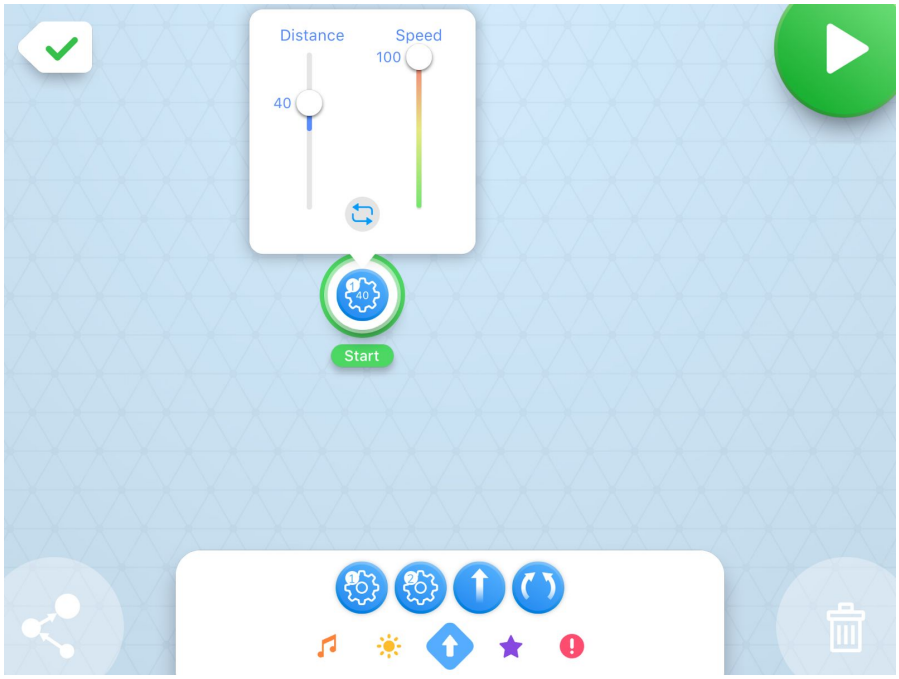
- Make a new project or continue programming in the same project.

1. Start with controlling only one motor - drag the **Motor 1** icon on the Programming Screen.



2. Set the **distance** and the **speed**. Try it out!

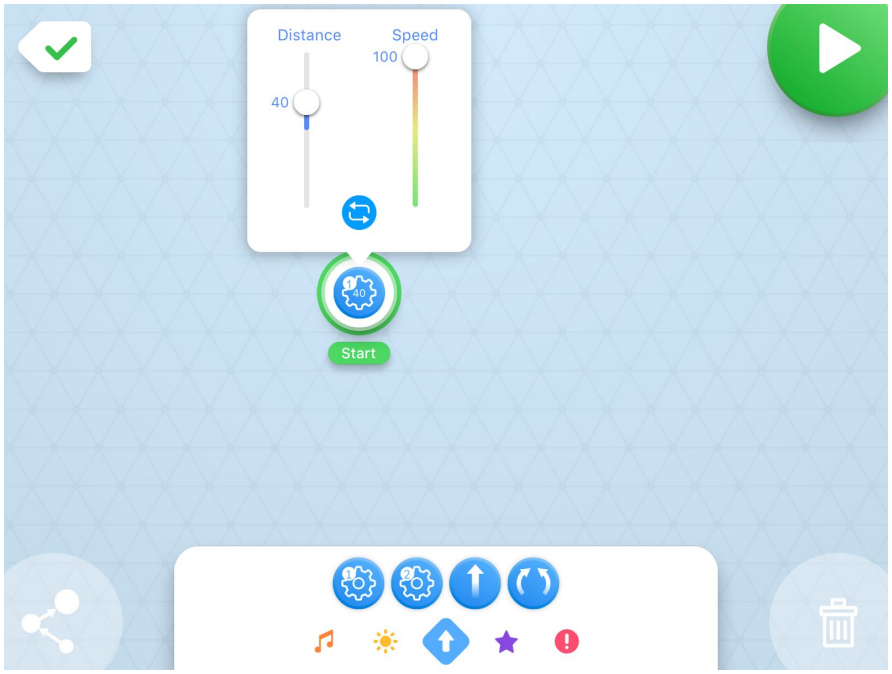


3. Make it drive in opposite side; you can use two options:

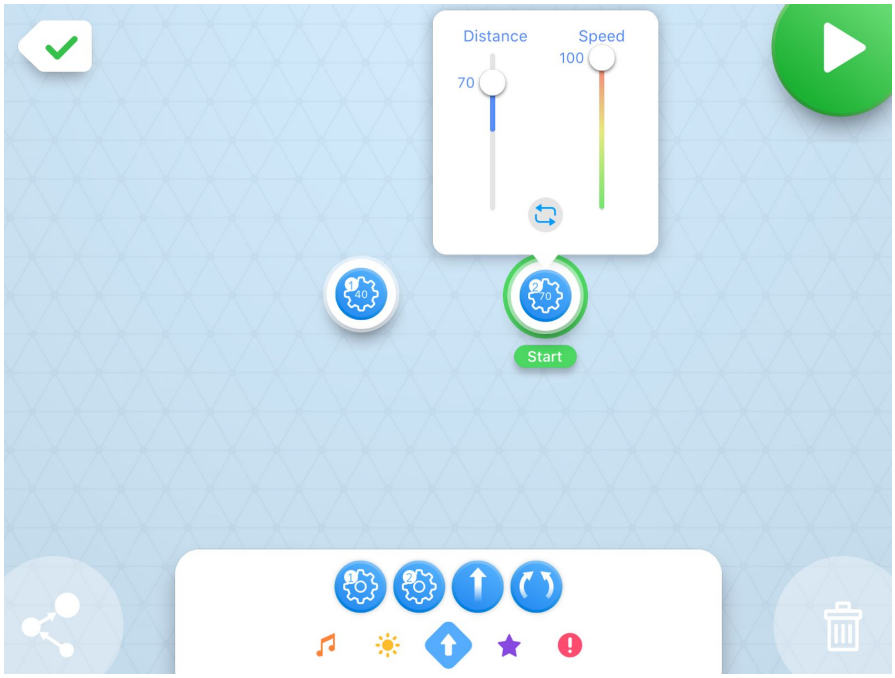
- Put the distance to minus

or

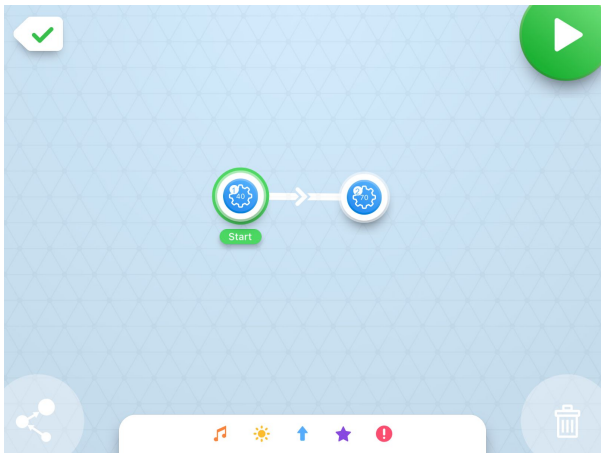
- Press the **Reverse Button**

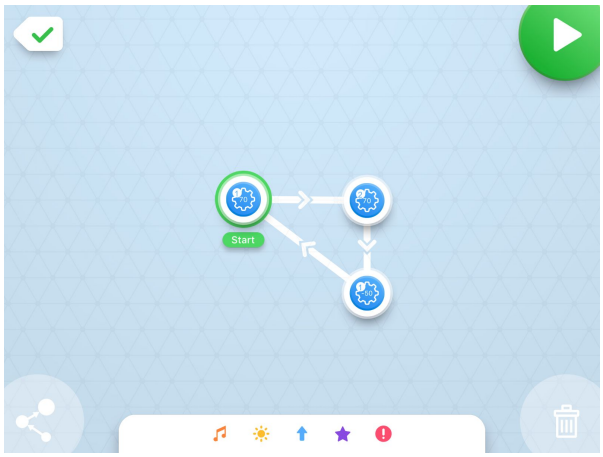


4. Add the Motor 2 to you code; set the **distance** and the **speed**. Try it independently - change the **Start Point.**

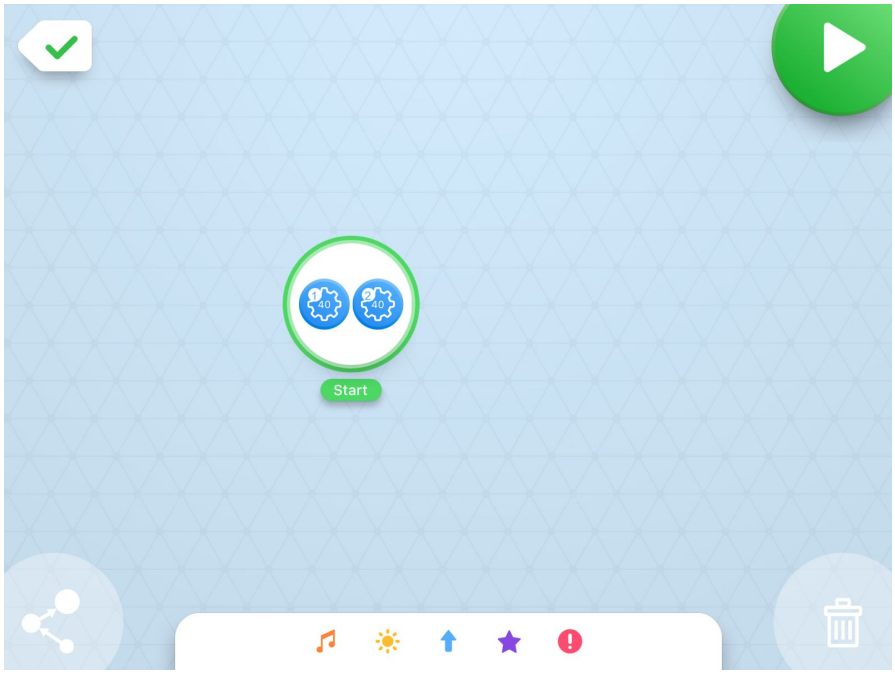


5. Connect **Motor 1** and **Motor 2** actions. Play around: try to change the direction of driving. Make a Loop.



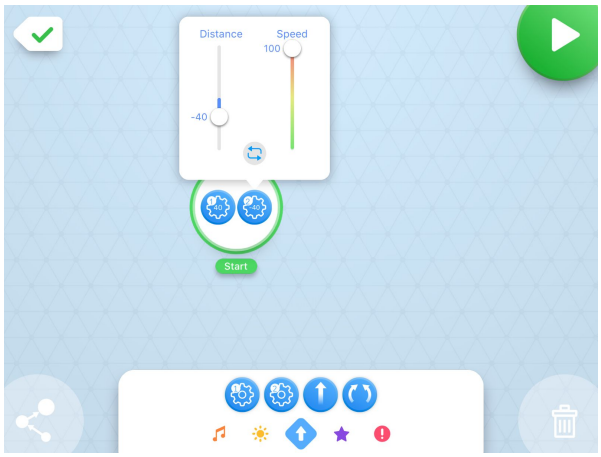


6. You can put the **Motor 1** and **Motor 2 Actions** in one **State**. It means that Actions will be played at the same time, parallel to one another, so the algorithm with **State** is called a **Parallel Code**.

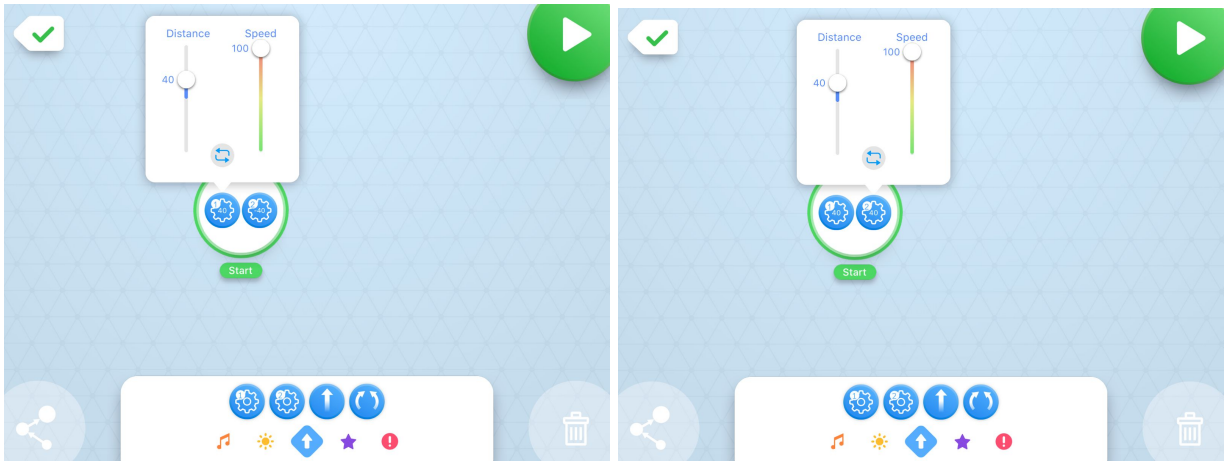


7. Pay attention to the **rotation direction of each DC Motor**:

- If you set up the different rotating direction for Motor 1 and Motor 2 (for example: Motor 1 = +40 and Motor 2 = -40) the Robo will drive forward;

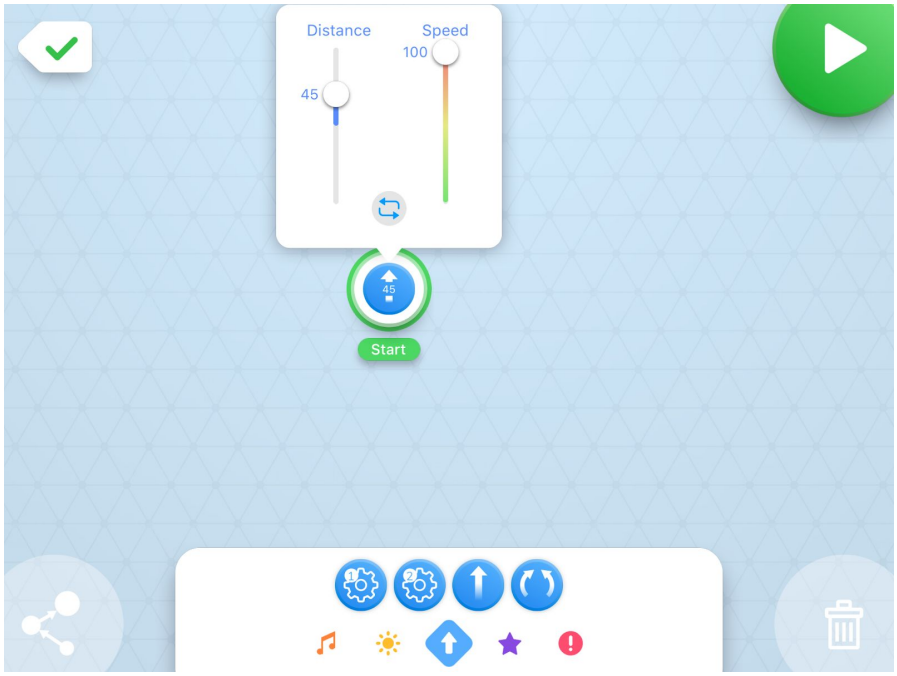


- If you set up the same direction for both Motor 1 and Motor 2 (for example: Motor 1 = Motor 2 = +40) - Robo will make a turn;

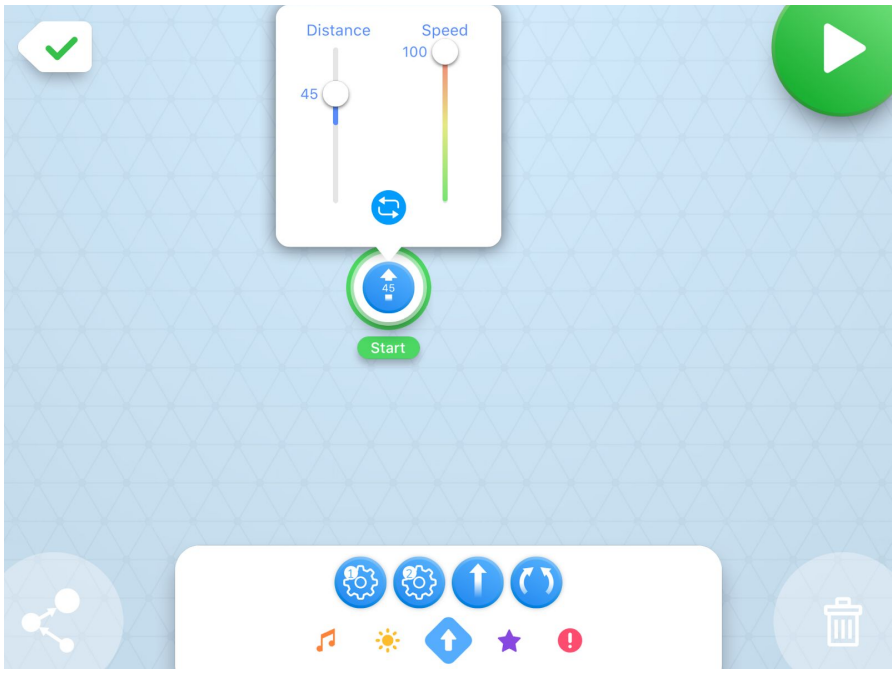


Explanation: Both DC Motors in the kit are technically the same and spin in the same direction. If you attach them to different sides of the Main Block, they are mirrored. In relation to the Main Block one of them will now spin forward and the other one backward, so the Robo will make a turn. To program the Robo drive forward you need to reverse one of the motors.

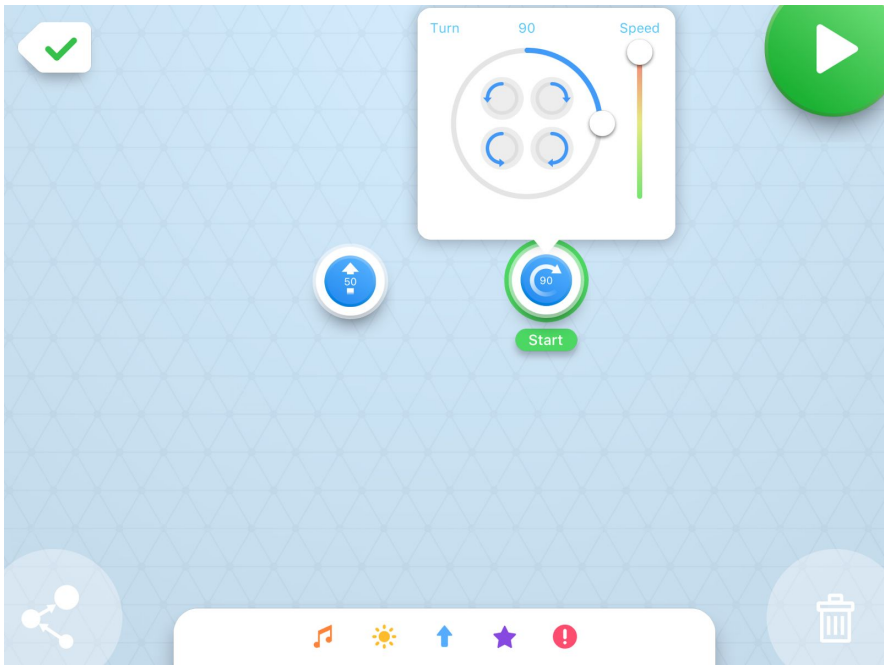
8. To program the Robo to drive forward you can also use **Drive Action**. Drag it on Programming Screen, set up the distance and speed. Try it out!



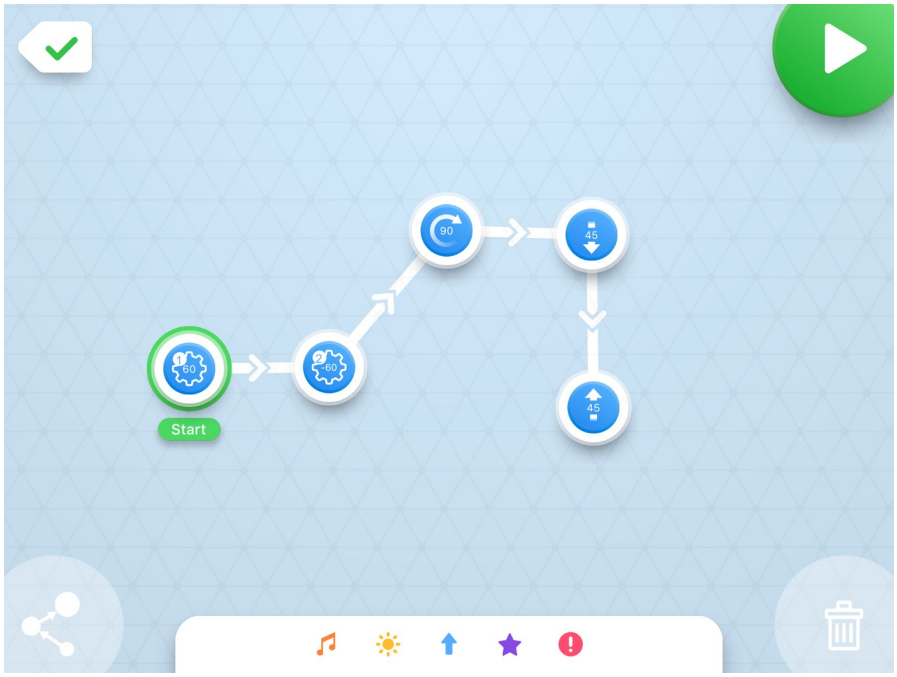
Attention: the DC Motors can be assembled in 2 different ways - Normal or Reverse. If they are in Reverse position, your Robo will drive backwards. In this situation, you need switch the mode in the settings from Normal to Reverse to drive forward.



9. To program the Robo to turn you can also use Turn Action. Drag it on Programming Screen, set up the angle and speed. Try it out!

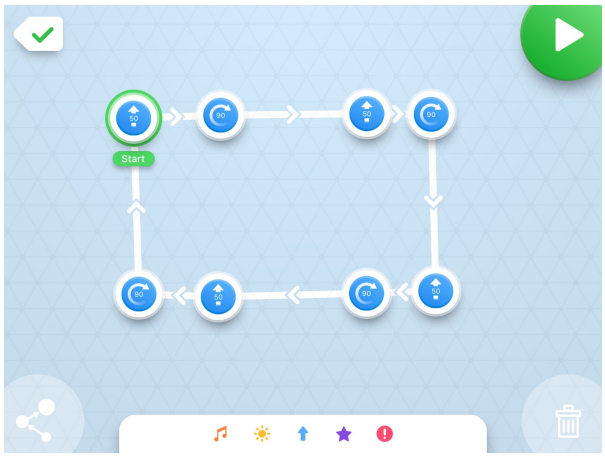


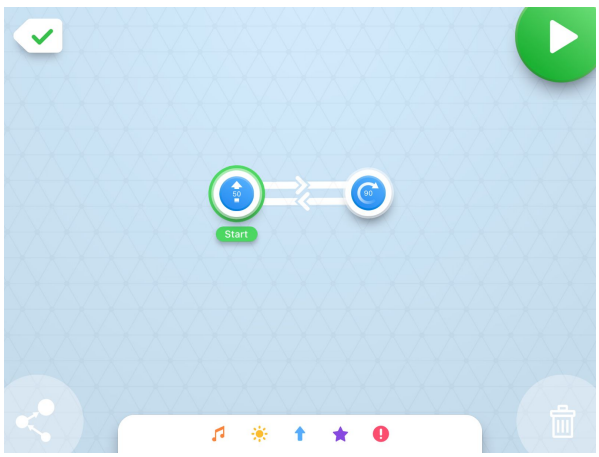
10. Play around: drag, set, and connect more Movement Actions or make a **Loop**! Use **Trash Bin** or change the **Starting Point** if you need! Pay attention to the **direction of arrows / transitions!**



11. Try to program your Robo to drive a square!

Here are two solutions for this task: the long code shows the logic and the short and more efficient code.

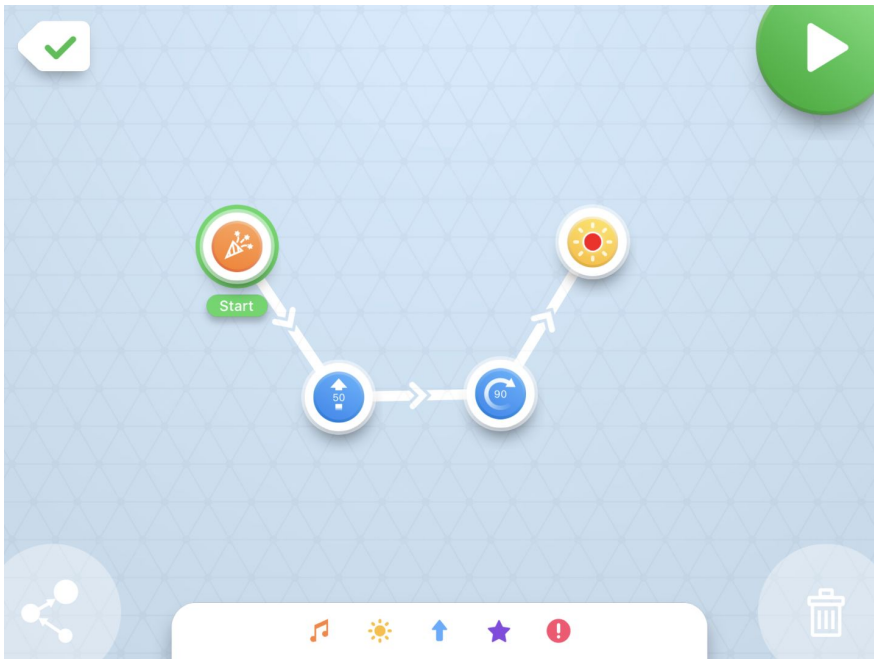




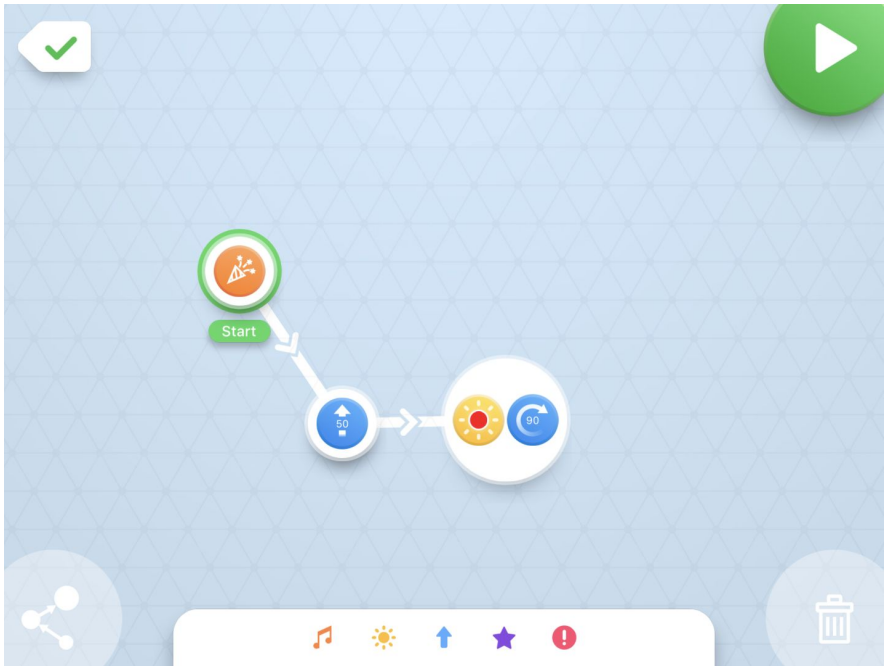
Step 4. All in one.

If the RGB Light and DC Motors are connected to the Main Block, you can use all already known actions: **Sound, Constant Light, Blink, Drive, Turn, Motor 1, and Motor 2**. But there are some tricks to do it in different ways.

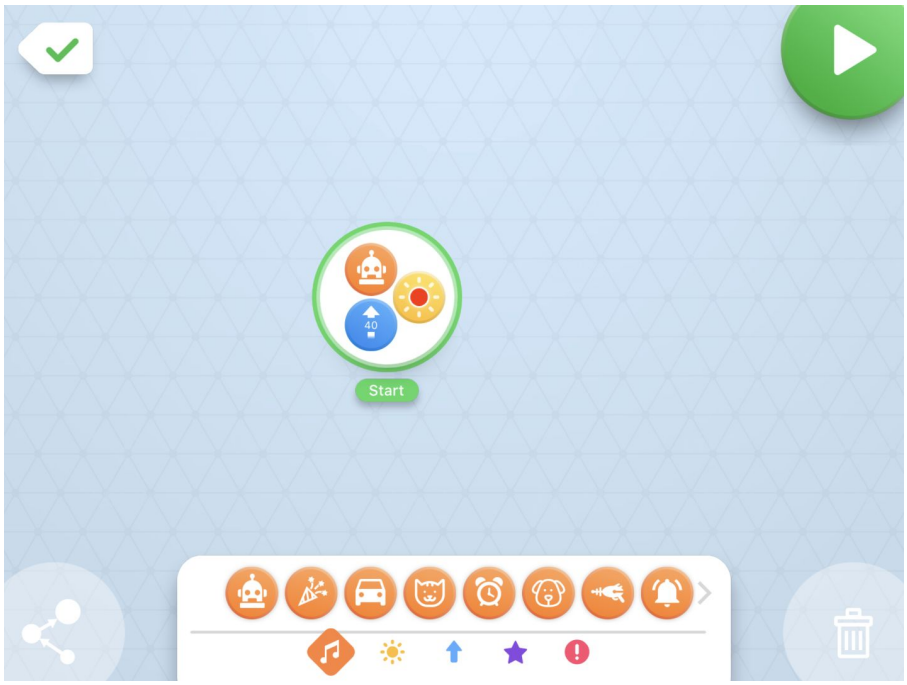
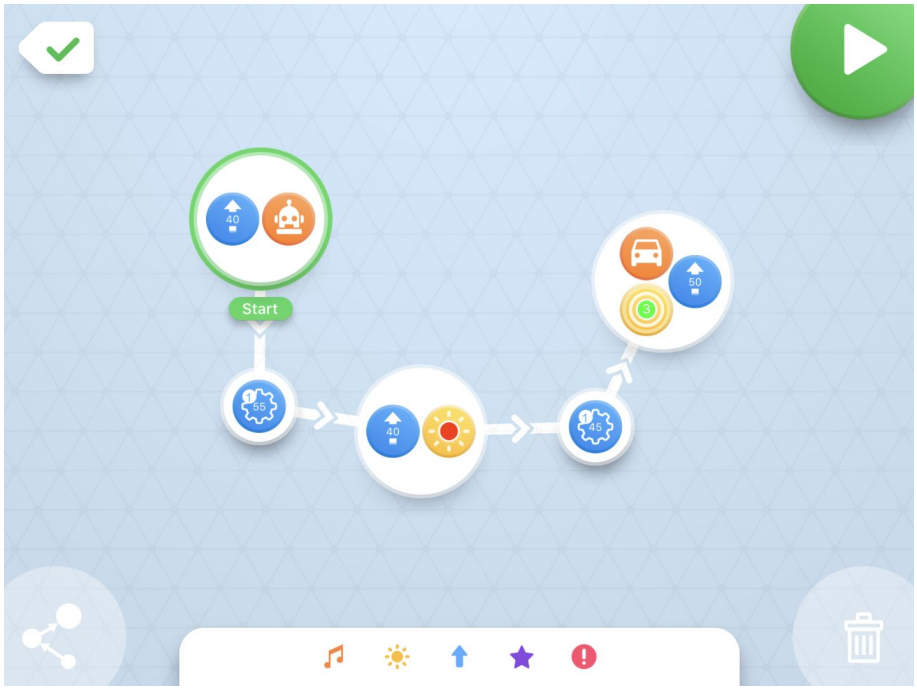
1. Drag all actions you want, set them, connect, and try! The Robo will reproduce actions one after another - this is called a **Sequential Code**.



2. Now let's make Robo do two actions in the same time. Drag one action on the other and they will combine into one big bubble, or **State**. Actions will be at the same time, parallel to one another, so the algorithm with State is called a **Parallel Code**.



Attention: actions have to be different types. Only one orange, one yellow, and one blue action can be put in the **State**; two actions of the same type (for example, two blue actions) cannot be put in the **State.**



3. Play around; make more common bubbles. You have now programmed the **Parallel Code**!